



Hojas de Estilo en Cascada, nivel 2 revisión 1 Especificación CSS2.1

Recomendación Del 25 De Febrero De 2004 Del Candidato de W3C

Este documento es una copia traducida de la Recomendación de W3C, sobre las Hojas de Estilo en Cascada, Nivel 2 Revisión 1 (CSS2.1).

Para realizar esta traducción me he basado en la existente de CSS2, realizada por: [Carlos Benavidez](#)

Este documento puede contener errores de traducción. Por favor si desea informar de errores encontrados en esta versión en castellano o tiene cualquier tipo de sugerencia, envíela a: crass2002@hotmail.com

Traducción: [SALISHAN](#)

Copyright © 1997-2003 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved.

Esta versión:

<http://www.w3.org/TR/2004/CR-CSS21-20040225>

La versión más reciente:

<http://www.w3.org/TR/CSS21>

Versión anterior:

<http://www.w3.org/TR/2003/WD-CSS21-20030915>

Redactores:

[Bert Bos](#) <bert@w3.org >

[Tantek Çelik](#) <tantekc@microsoft.com >

[Ian Hickson](#) <@hixie.ch ian >

[Håkon Wium Lie](#) <howcome@opera.com > de Håkon Wium

Este documento está también disponible en estos formatos no-normativos: texto plano, archivo gzip'ed tar, zip file, gzip'ed PostScript, PDF. Vea las [traducciones](#).

Copyright © 2004 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#) and [software licensing](#) rules apply.

Resumen

Esta especificación define las Hojas de Estilo en Cascada (Cascading Style Sheets), nivel 2 revisión 1 (CSS 2.1). CSS 2.1 es un lenguaje de hoja de estilo que permite que los autores y los usuarios asocien un estilo (por ejemplo, las fuentes y espaciado) a los documentos estructurados (por ejemplo, documentos HTML y aplicaciones XML). Separando el estilo de presentación del contenido de los documentos, CSS 2.1 simplifica la creación y mantenimiento de los sitios Web.

CSS 2.1 está basado en CSS2 [[CSS2](#)] el cuál se cimienta en CSS1 [[CSS1](#)]. Este brinda soporte específico a hojas de estilo específicas para cada medio, de modo que los autores puedan adaptar la presentación de sus documentos a los navegadores (browsers) visuales, a los dispositivos sonoros, a las impresoras, a los dispositivos de braille, de mano, etc. También soporta el posicionamiento de contenidos, disposición de tablas, aspectos para la internacionalización, y algunas características relacionadas con la interfaz del usuario.

CSS 2.1 corrige algunos errores de CSS2 (el más importante es una nueva definición de los altos/anchos (height/width) de elementos de posiciones absolutas, más influencia para la cualidad del "estilo" de los HTML y de un nuevo cálculo de las propiedades de 'clip'), y agrega algunas propiedades altamente solicitadas que se han puesto en práctica ya extensamente. Pero sobretodo el CSS 2.1 representa una "imagen" de lo usado de CSS: este consiste en todas las propiedades de CSS que se han puesto en práctica desde la fecha de la publicación de la recomendación.

Estado de este documento

Esta sección describe el estado de este documento a la hora de su publicación. Otros documentos pueden reemplazarlo. Una lista de las publicaciones actuales de W3C y de la revisión más actual de este informe técnico se puede encontrar en [el índice técnico de los informes de W3C en http://www.w3.org/TR/](#).

Éste es un candidato recomendado de W3C, lo que significa que la especificación se ha repasado extensamente y W3C recomienda que este sea puesto en práctica. Seguirá siendo candidato recomendado [por lo menos hasta el 1 de septiembre de 2004](#).

La publicación como candidato recomendado no implica el endoso de los miembros de W3C. Es inadecuado citar este documento como otro trabajo en proceso

Se proporcionará una herramienta de testeo y un informe de la puesta en práctica antes de que el documento sea una recomendación propuesta.

La lista ([archivada](#)) publica de correos www-style@w3.org (ver [instrucciones](#)) es preferible para la discusión de esta y otras especificación del área de Estilos. Para comentar algo al respecto a este documento, ponga por favor el texto "CSS21" en asunto tema, preferiblemente como esto: "[CSS21] < resumen del comentario >"

Este documento es elaborado por [el grupo de trabajo de CSS](#) (la parte [de la actividad del estilo](#), ver [resumen](#)).

Los accesos de la patente relevantes al CSS se pueden encontrar en la página pública del acceso [de la patente del grupo de trabajo](#).

Criterios De la Salida De la Recomendación Del Candidato

Para que esta especificación salga de la etapa del RC, las condiciones siguientes deben ser resueltas:

1. Debe haber por lo menos dos implementaciones interoperables para cada propiedad. Para los propósitos de este criterio, definimos los siguientes términos:

propiedad

Una sección o una subsección de la especificación.

interoperable

testear los respectivos casos en las herramientas de testeo, o, si la implementación no es un navegador web, realizar pruebas equivalentes. Cada prueba relevante en las herramientas de testeo debe tener una prueba equivalente creada si una AU (aplicación del usuario) es utilizada para demandar interoperabilidad. Además si una AU (aplicación del usuario) es utilizada para demandar interoperabilidad, después debe haber una o más AUs (aplicaciones de usuarios) adicionales que también puedan pasar esas pruebas equivalentes de la misma manera para alcanzar el objetivo de la interoperabilidad. Las pruebas equivalentes se deben hacer públicas y disponibles para los propósitos de la revisión.

Implementación

un agente del usuario que:

1. pone la propiedad en ejecución.
 2. está disponible (es decir público descargable o disponible en otro punto público del mecanismo de la venta). Ésta es el requisito de "demostración" requisito.
 3. es enviado (Ej. desarrollos, privados o versiones no oficiales son escasos).
 4. no es experimental (es decir, se piensa para una audiencia grande y podría ser utilizado sobre una base diaria).
2. Debe haber transcurrido un mínimo de seis meses del período de RC. Éste debe asegurarse de que ha dado bastante tiempo para la aparición de cualquier error importante.
 3. El período del RC será extendido si las implementaciones son lentas en aparecer.
 4. Las propiedades que no estaban en CSS1 serán retiradas (así se reduce lista de "todas las" propiedades mencionadas arriba) si dos o mas implementaciones interoperables de esas propiedades no se encuentran para el final del período del RC.
 5. Las propiedades también serán retiradas si las pruebas suficientes y adecuadas (por el juicio del grupo de trabajo) no se han producido para esas propiedades para el final del período del RC.

Contenido Abreviado

- [1 Acerca de la Especificación CSS 2.1](#)
- [2 Introducción a CSS 2.1](#)
- [3 Conformidad: Requisitos y Recomendaciones](#)
- [4 Sintaxis y tipos de datos básicos](#)
- [5 Selectores](#)
- [6 Asignación de valores a las propiedades, Cascada y Herencia](#)
- [7 Tipos de medios](#)
- [8 Modelo de caja](#)
- [9 Modelo del formato visual](#)
- [10 Detalles del modelo de formato visual](#)
- [11 Efectos visuales](#)
- [12 Contenido generado, numeración automática y listas](#)
- [13 Medios paginados](#)
- [14 Colores y fondos](#)
- [15 Fuentes](#)
- [16 Texto](#)
- [17 Tablas](#)
- [18 Interfaz del usuario](#)
- [Apéndice A: Hojas estilo de auditivas](#)
- [Apéndice B: Bibliografía](#)
- [Apéndice C: Cambios](#)
- [Apéndice D: Hoja de estilo por defecto para HTML 4,0](#)
- [Apéndice E: Descripción de elaboración de contextos apilados](#)
- [Apéndice F: Tabla completa de las propiedades](#)
- [Apéndice G: Gramática de CSS 2.1](#)
- [Apéndice I: Índice](#)

Contenido completo

- **1 Acerca de la especificación CSS 2.1**
 - 1.1 Css 2.1 contra CSS 2
 - 1.2 Lectura de la especificación
 - 1.3 Cómo se organiza la especificación
 - 1.4 Convenciones
 - 1.4.1 Elementos y atributos en el lenguaje del documento
 - 1.4.2 Definiciones de las propiedades de CSS
 - Valor
 - Inicial
 - Se aplica a
 - Se hereda
 - Porcentaje
 - Grupos de medios
 - Valor computado
 - 1.4.3 Propiedades resumidas
 - 1.4.4 Notas y ejemplos
 - 1.4.5 Imágenes y descripciones largas
 - 1.5 Reconocimientos
 - 1.6 Aviso De Copyright
- **2 Introducción a CSS 2.1**
 - 2.1 Breve guía de CSS 2.1 para HTML
 - 2.2 Breve guía de CSS 2.1 para XML
 - 2.3 El modelo de proceso del CSS 2.1
 - 2.3.1 El lienzo
 - 2.3.2 Modelo de destinatarios de CSS 2.1
 - 2.4 Principios del diseño del CSS
- **3 Conformidad: Requisitos y recomendaciones**
 - 3.1 Definiciones
 - 3.2 Conformidad
 - 3.3 Condiciones de error
 - 3.4 Tipo de contenido text/css
- **4 Sintaxis y tipos de datos básicos**
 - 4.1 Sintaxis
 - 4.1.1 Comandos
 - 4.1.2 Palabras claves
 - Extensiones específicas-navegador
 - Notas Históricas Informativas
 - 4.1.3 Caracteres y mayúsculas/minúsculas
 - 4.1.4 Declaraciones
 - 4.1.5 Reglas-arroba
 - 4.1.6 Bloques
 - 4.1.7 Sistemas de reglas, bloques de declaraciones, y selectores
 - 4.1.8 Declaraciones y propiedades
 - 4.1.9 Comentarios
 - 4.2 Reglas para manejar errores de análisis
 - 4.3 Valores
 - 4.3.1 Números enteros y números reales
 - 4.3.2 Medidas

- [4.3.3 Porcentajes](#)
 - [4.3.4 URL + URN = URI](#)
 - [4.3.5 Contadores](#)
 - [4.3.6 Colores](#)
 - [4.3.7 Cadenas](#)
 - [4.3.8 Valores no soportados](#)
 - [4.4 Representación de los documentos CSS](#)
 - [4.4.1 Referencias a los caracteres no representados en una codificación de caracteres](#)
- **[5 selectores](#)**
 - [5.1 Equivalencia de patrones](#)
 - [5.2 Sintaxis del selector](#)
 - [5.2.1 Agrupamiento](#)
 - [5.3 Selector universal](#)
 - [5.4 Selectores de tipos](#)
 - [5.5 Selectores de descendiente](#)
 - [5.6 Selectores de hijos](#)
 - [5.7 Selectores de hermanos adyacentes](#)
 - [5.8 Selectores de atributos](#)
 - [5.8.1 Atributos equivalentes y valores de atributos](#)
 - [5.8.2 Valores predeterminados de los atributos en la DTDs](#)
 - [5.8.3 Selectores de clase](#)
 - [5.9 Selectores ID](#)
 - [5.10 Pseudo-elementos y pseudo-clases](#)
 - [5.11 Pseudo-clases](#)
 - [5.11.1 La pseudo-clase :first-child](#)
 - [5.11.2 Las pseudo-clases vínculo :link y :visited](#)
 - [5.11.3 Las pseudo-clases dinámicas: :hover :active y :focus](#)
 - [5.11.4 La pseudo-clase del lenguaje: :lang](#)
 - [5.12 Pseudo-elementos](#)
 - [5.12.1 El pseudo-elemento :first-line](#)
 - [5.12.2 El pseudo-elemento :first-letter](#)
 - [5.12.3 Los pseudo-elementos :before y :after](#)
- **[6 Asignación de valores a las propiedades, Cascada, y Herencia](#)**
 - [6.1 Valores especificados, computados, y reales](#)
 - [6.1.1 Valores especificados](#)
 - [6.1.2 Valores computados](#)
 - [6.1.3 Valores usados](#)
 - [6.1.4 Valores reales](#)
 - [6.2 Herencia](#)
 - [6.2.1 El valor 'inherit'](#)
 - [6.3 La regla @import](#)
 - [6.4 Cascada](#)
 - [6.4.1 Orden de conexión en cascada](#)
 - [6.4.2 Las reglas !important](#)
 - [6.4.3 Cálculo de especificidad de un selector](#)
 - [6.4.4 Precedencia de las indicaciones de presentación fuera de CSS](#)
- **[7 Tipos de medios](#)**
 - [7.1 Introducción a los tipos de medios](#)
 - [7.2 Especificar las hojas de estilo dependientes de los medios](#)

- [7.2.1 La regla del @media](#)
 - [7.3 Tipos de medios reconocidos](#)
 - [7.3.1 Grupos de medios](#)
- **[8 Modelo de caja](#)**
 - [8.1 Dimensiones de la caja](#)
 - [8.2 Ejemplo de márgenes, rellenos y bordes](#)
 - [8.3 propiedades del margen: 'margen-top' 'margen-right' 'margen-bottom' 'margen-left' y 'margin'](#)
 - [8.3.1 Márgenes cerrados](#)
 - [8.4 Propiedades de relleno: 'padding-top' 'padding-right' 'padding-bottom' 'padding-left' y 'padding'](#)
 - [8.5 Propiedades del borde](#)
 - [8.5.1 Ancho del borde: 'border-top-width' 'border-right-width' 'border-bottom-width' 'border-left-width' y 'border-width'](#)
 - [8.5.2 Color del borde: 'border-top-color' 'border-right-color' 'border-bottom-color' 'border-left-color' y 'border-color'](#)
 - [8.5.3 Estilo del borde: 'border-top-style' 'border-right-style' 'border-bottom-style' 'border-left-style' y 'border-style'](#)
 - [8.5.4 propiedades resumidas del borde: 'border-top' 'border-left' 'border-bottom' 'border-right' y 'border'](#)
 - [8.6 El modelo de caja para los elementos en línea en contexto bidireccional](#)
- **[9 Modelo de formato visual](#)**
 - [9.1 Introducción al modelo de formato visual](#)
 - [9.1.1 El acceso visual](#)
 - [9.1.2 bloques de contención](#)
 - [9.2 Control de la generación de cajas](#)
 - [9.2.1 Elementos a nivel de bloque y cajas de bloque](#)
 - [Cajas de bloque anónimas](#)
 - [9.2.2 Elementos a nivel de línea y cajas en línea](#)
 - [Cajas en línea anónimas](#)
 - [9.2.3 Cajas run-in](#)
 - [9.2.4 La propiedad 'display'](#)
 - [9.3 Esquemas de posicionamiento](#)
 - [9.3.1 Elección de un de posicionamiento: la propiedad 'position'](#)
 - [9.3.2 Desplazamiento de las cajas 'top', 'right', 'bottom', 'left'](#)
 - [9.4 Flujo normal](#)
 - [9.4.1 Contexto del formato de bloque](#)
 - [9.4.2 Contexto del formato en línea](#)
 - [9.4.3 Posicionamiento relativo](#)
 - [9.5 Flotantes](#)
 - [9.5.1 Posicionamiento del flotante: la propiedad 'float'](#)
 - [9.5.2 Control del flujo al costado del flotante: 'clear'](#)
 - [9.6 Posicionamiento absoluto](#)
 - [9.6.1 Posicionamiento fijo](#)
 - [9.7 Relaciones entre 'display', 'position', y 'float'](#)
 - [9.8 Comparación entre flujo normal, flotantes, y posicionamiento absoluto](#)
 - [9.8.1 Flujo normal](#)
 - [9.8.2 Posicionamiento relativo](#)
 - [9.8.3 Caja flotante](#)
 - [9.8.4 Posicionamiento absoluto](#)

- [9.9 Presentación por capas](#)
 - [9.9.1 Especificar el nivel de apilado: la propiedad 'z-index'](#)
- [9.10 Dirección del texto: las propiedades 'direction' y 'unicode-bidi'](#)
- **[10 Detalles del modelo de formato visual](#)**
 - [10.1 Definición de "bloque de contención"](#)
 - [10.2 Ancho del contenido: la propiedad 'width'](#)
 - [10.3 Computando anchos y márgenes](#)
 - [10.3.1 Elementos no reemplazados, en línea](#)
 - [10.3.2 Elementos reemplazados, en línea](#)
 - [10.3.3 Elementos no reemplazados en flujo normal, a nivel de bloque](#)
 - [10.3.4 Elementos reemplazados en flujo normal, a nivel de bloque](#)
 - [10.3.5 Elementos no reemplazados, flotantes](#)
 - [10.3.6 Elementos reemplazados, flotantes](#)
 - [10.3.7 Elementos no reemplazados, posicionados absolutamente](#)
 - [10.3.8 Elementos reemplazados, posicionados absolutamente](#)
 - [10.3.9 Elementos no reemplazados en flujo normal, de 'bloques en línea'](#)
 - [10.3.10 Elementos reemplazados en flujo normal, de 'bloques en línea'](#)
 - [10.4 Anchos mínimos y máximos: 'min-width' y 'max-width'](#)
 - [10.5 Altura del contenido: la propiedad 'height'](#)
 - [10.6 Computando alturas y márgenes](#)
 - [10.6.1 Elementos no reemplazados, en línea](#)
 - [10.6.2 Elementos reemplazados, en línea, elementos reemplazados en flujo normal a nivel de bloque, elementos reemplazados en flujo normal de bloques en línea y elementos reemplazados flotantes](#)
 - [10.6.3 Elementos no reemplazados en flujo normal a nivel de bloque cuando 'overflow' es computado como 'visible'](#)
 - [10.6.4 Elementos no reemplazados, posicionados absolutamente](#)
 - [10.6.5 Elementos reemplazados, posicionados absolutamente](#)
 - [10.6.6 Elementos no reemplazados en flujo normal a nivel de bloque, cuando 'overflow' no es computado como 'visible', elementos no reemplazados de bloques en línea, y elementos no reemplazados flotantes](#)
 - [10.6.7 Alturas 'auto' para el contexto de formato de bloque raíz](#)
 - [10.7 Alturas mínimas y máximas: 'min-height' y 'max-height'](#)
 - [10.8 Cálculo de la altura de la línea: las propiedades 'line-height' y 'vertical-height'](#)
 - [10.8.1 Interlineado y medio interlineado](#)
- **[11 Efectos visuales](#)**
 - [11.1 Desbordamiento y recorte](#)
 - [11.1.1 Desbordamiento: la propiedad 'overflow'](#)
 - [11.1.2 Recorte: la propiedad 'clip'](#)
 - [11.2 Visibilidad: la propiedad 'visibility'](#)
- **[12 Contenido generado, numeración automática y listas](#)**
 - [12.1 Los pseudo-elementos :before y :after](#)
 - [12.2 La propiedad 'content'](#)
 - [12.3 Comillas](#)
 - [12.3.1 Especificación de comillas con la propiedad 'quotes'](#)
 - [12.3.2 Inserción de comillas con la propiedad 'content'](#)
 - [12.4 Contadores y numeración automática](#)
 - [12.4.1 Contadores anidados y área de alcance](#)
 - [12.4.2 Estilos de contadores](#)
 - [12.4.3 Contadores en elementos con 'display: none'](#)

- [12.5 Listas](#)
 - [12.5.1 Listas: las propiedades 'list-style-type', 'list-style-image', 'list-style-position' y 'list-style'](#)
- **[13 Medios paginados](#)**
 - [13.1 Introducción a los medios paginados](#)
 - [13.2 Cajas de página: la regla @page](#)
 - [13.2.1 Márgenes de la página](#)
 - [Procesando las cajas de página que no ajustan a una hoja de destino](#)
 - [Ubicación de la caja de página en la hoja](#)
 - [13.2.2 Selectores de página: seleccionando paginas izquierda, derecha, y primera](#)
 - [13.2.3 Contenido fuera de caja de la página](#)
 - [13.3 Saltos de página](#)
 - [13.3.1 Propiedades de salto de página: 'page-break-before', 'page-break-after' 'page-break-inside'](#)
 - [13.3.2 Saltos dentro de los elementos: 'orphans' y 'widows'](#)
 - [13.3.3 Saltos de páginas permitidos](#)
 - [13.3.4 Saltos de páginas forzados](#)
 - [13.3.5 Los "mejores" saltos de página](#)
 - [13.4 Cascada en el contexto de la página](#)
- **[14 Colores y fondos](#)**
 - [14.1 Color del primer plano: la propiedad 'color'](#)
 - [14.2 El fondo](#)
 - [14.2.1 Propiedades del fondo: 'background-color', 'background-image', 'background-repeat', 'background-attachment' 'background-position' y 'background'](#)
 - [14.3 Corrección gamma](#)
- **[15 Fuentes](#)**
 - [15.1 Introducción](#)
 - [15.2 Algoritmo de equivalencia de fuentes](#)
 - [15.3 Familias de fuentes: la propiedad 'font-family'](#)
 - [15.4 Estilos de la fuente: la propiedad 'font-style'](#)
 - [15.5 Small-caps: la propiedad 'font-variant'](#)
 - [15.6 Grosor de la fuente: la propiedad 'font-weight'](#)
 - [15.7 Tamaño de fuente: la propiedad 'font-size'](#)
 - [15.8 Propiedad abreviada de fuente: la propiedad 'font'](#)
- **[16 Text](#)**
 - [16.1 Sangría: la propiedad 'text-indent'](#)
 - [16.2 Alineación: la propiedad 'text-align'](#)
 - [16.3 Decoración](#)
 - [16.3.1 Subrayado, sobrerayado, tachado, y parpadeo: la propiedad 'text-decoration'](#)
 - [16.4 Espaciado de letras y palabras: las propiedades 'letter-spacing' y 'word-spacing'](#)
 - [16.5 Capitalización: la propiedad 'text-transform'](#)
 - [16.6 Espacios en blanco: la propiedad 'white-space'](#)
 - [16.6.1 Modelo de procesado de 'white-space'](#)
 - [16.6.2 Ejemplo de bidireccionalidad con white-space cerrados](#)

- **17 Tablas**

- [17.1 Introducción a las tablas](#)
- [17.2 El modelo de tabla de CSS](#)
 - [17.2.1 Objetos anónimos de la tabla](#)
- [17.3 Columnas](#)
- [17.4 Tablas en el modelo de formato visual](#)
 - [17.4.1 Posición y alineación del encabezado](#)
- [17.5 Composición visual del contenido de la tabla](#)
 - [17.5.1 Capas y transparencia en la tabla](#)
 - [17.5.2 Algoritmos para el ancho de la tabla: la propiedad 'table-layout'](#)
 - [Composición fija de la tabla](#)
 - [Composición automática de la tabla](#)
 - [17.5.3 Algoritmos para la altura de la tabla](#)
 - [17.5.4 Alineación horizontal en una columna](#)
 - [17.5.5 Efectos dinámicos en filas y columnas](#)
- [17.6 Bordas](#)
 - [17.6.1 El modelo de bordes separado](#)
 - [Bordas y fondos alrededor de las celdas vacías: la propiedad 'empty-cells'](#)
 - [17.6.2 El modelo de bordes cerrados](#)
 - [Resolución de conflictos de bordes](#)
 - [17.6.3 Estilos de bordes](#)

- **18 Interfaz de usuario**

- [18.1 Cursores: la propiedad 'cursor'](#)
- [18.2 Colores del Sistema CSS2](#)
- [18.3 Preferencias de fuentes del usuario](#)
- [18.4 Contornos dinámicos: la propiedad 'outline'](#)
 - [18.4.1 Contornos y foco](#)
 - [18.5 Ampliación](#)

- **Apéndice A. Hojas de estilo auditivas**

- [A.1 Tipos de medios 'auditivos' y 'voz'](#)
- [A.2 Introducción a las hojas de estilos auditivas](#)
 - [A.2.1 Ángulos](#)
 - [A.2.2 Tiempos](#)
 - [A.2.3 Frecuencias](#)
- [A.3 Propiedades del volumen: 'volume'](#)
- [A.4 Propiedades del habla: 'speak'](#)
- [A.5 Propiedades de la pausa: 'pause-before', 'pause-after' y 'pause'](#)
- [A.6 Propiedades de la señal: 'cue-before', 'cue-after' y 'cue'](#)
- [A.7 Propiedades de la mezcla: 'play-during'](#)
- [A.8 Propiedades espaciales: 'azimuth' y 'elevation'](#)
- [A.9 Propiedades de las características de la voz: 'speech-rate' 'voz-family' 'pitch' 'pitch-range' 'stress' y 'richness'](#)
- [A.10 Propiedades de lectura: 'speak-punctuation' y 'speak-numeral'](#)
- [A.11 Procesado de tablas de audio](#)
 - [JA.11.1 Cabeceras de la voz: la propiedad 'speak-header'](#)
- [A.12 Ejemplos de hojas de estilo para HTML](#)
- [A.13 Emacspeak](#)

- **Apéndice B. Bibliografía**
 - B.1 Referencias normativas
 - B.2 Referencias informativas
- **Apéndice C. Cambios**
 - C.1 Valores de propiedades adicionales
 - C.1.1 Sección 4.3.5 Colores
 - C.1.2 Sección 9.2.4 la propiedad de la 'display'
 - C.1.3 Sección 12.2 la propiedad 'content'
 - C.1.4 Sección 18.1 Cursores: la propiedad del 'cursor'
 - C.1.5 Sección 16.6 Espacios en blanco: la propiedad 'white-space'
 - C.2 Cambios
 - C.2.1 Sección 3.2 Conformidad
 - C.2.2 Sección 6.1.2 Valores computados
 - C.2.3 Sección 6.4.3 Calculo de especificidad de un selector
 - C.2.4 Sección 6.4.4 Precedencia de las indicaciones de presentación fuera de CSS
 - C.2.5 Capítulo 9 Modelo de formato visual
 - C.2.6 Sección 10.3.7 Posicionamiento absoluto, elementos no reemplazados
 - C.2.7 Sección 10.6.4 Posicionamiento absoluto, elementos no reemplazados
 - C.2.8 Sección 11.1.2 Recorte: la propiedad 'clip'
 - C.2.9 Sección 14.2.1 Propiedades del fondo
 - C.2.10 Sección 17.4.1 Posicionamiento y alineamiento del título
 - C.2.11 Sección 17.5.4 Alineación horizontal en una columna
 - C.2.12 Sección 17.6 Bordes
 - C.2.13 Capítulo 12 Contenido generado, numeración automática y listas
 - C.2.14 Sección 12.2 La propiedad 'content'
 - C.2.15 Capítulo 13 Medios paginados
 - C.2.16 Capítulo 15 Fuentes
 - C.2.17 Capítulo 16 Texto
 - C.2.18 Apéndice A Hojas de estilos auditivas
 - C.2.19 Otros
 - C.3 Errores
 - C.3.1 Propiedades abreviadas
 - C.3.2.4 Sección 4.1.1 (y G2)
 - C.3.3 Sección 4.1.3 Caracteres y caso
 - C.3.4 Sección 4.3 (problema doble de signos)
 - C.3.5 Sección 4.3.2 Medidas
 - C.3.6 Sección 4.3.6
 - C.3.7 Sección 5.10 Pseudo-elementos y pseudo-clases
 - C.3.8.8 Sección 8.2 Ejemplos de márgenes, rellenos, y bordes
 - C.3.9 Sección 8.5.2 Color del borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', y 'border-color'
 - C.3.10 Sección 8.4 Propiedades de relleno
 - C.3.11 Sección 8.5.3 Estilo del borde
 - C.3.12 Sección 8.5.4 Propiedades abreviadas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'
 - C.3.13 Sección 8.5.4 Propiedades abreviadas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'
 - C.3.14 Sección 9.3.1
 - C.3.15 Sección 9.3.2

- [C.3.16 Sección 9.4.3](#)
- [C.3.17 Sección 9.7 Relaciones entre 'display', 'position', y 'float'](#)
- [C.3.18 Sección 10.3.2 Elementos en línea, reemplazados \(y 10.3.4, 10.3.6, y 10.3.8\)](#)
- [C.3.19 Sección 10.3.3](#)
- [C.3.20 Sección 10.6.2 Elementos en línea, reemplazados... \(y 10.6.5\)](#)
- [C.3.21 Sección 10.6.3](#)
- [C.3.22 Sección 11.1.1](#)
- [C.3.23 Visibilidad: la propiedad 'visibility'](#)
- [C.3.24 Sección 12.6.2 Listas](#)
- [C.3.25 Sección 15.2.6](#)
- [C.3.26 Sección 15.5](#)
- [C.3.27 Sección 16.6 Espacios en blanco: la propiedad 'white-space'](#)
- [C.3.28 Sección 17.2 El modelo de la tabla de CSS](#)
- [C.3.29 Sección 17.2.1 Objetos anónimos de la tabla](#)
- [C.3.30 Sección 17.5 Composición visual del contenido de las tablas](#)
- [C.3.31 Sección 17.5 Composición visual del contenido de las tablas](#)
- [C.3.32 Sección 17.5.1 Capas y transparencia de la tabla](#)
- [C.3.33 Sección 17.6.1 El modelo de bordes separados](#)
- [C.3.34 Apéndice D.2 Escáner léxico](#)
- [C.4 Aclaraciones](#)
 - [C.1.1 Sección 2.2 Breve tutorial de CSS2 para XML](#)
 - [C.4.2 Sección 4.111](#)
 - [C.4.3 sección 5.5](#)
 - [C.4.4 Sección 5.9 Selectores de la ID](#)
 - [C.4.5 Sección 5.12.1 Pseudo-elemento :first-line](#)
 - [C.4.6 Sección 6.2.1](#)
 - [C.4.7 Sección 6.4 Cascada](#)
 - [C.4.8 Sección 6.4.3 Calculo de especificidad de un selector](#)
 - [C.4.9 Sección 7.3 Tipos de medios reconocidos](#)
 - [C.4.10 Sección 8.1](#)
 - [C.4.11 Sección 8.3.1](#)
 - [C.4.12 Sección 9.4.2](#)
 - [C.4.13 Sección 9.4.3](#)
 - [C.4.14 Sección 9.10](#)
 - [C.4.15 Block-level, elementos no reemplazados en flujo normal](#)
 - [C.4.16 Sección 10.5 Altura del contenido: la propiedad 'height'](#)
 - [C.4.17 Sección 10.8.1](#)
 - [C.4.18 Sección 11.1](#)
 - [C.4.19 Sección 11.1.1](#)
 - [C.4.20 Sección 11.1.2](#)
 - [C.4.21 Sección 12.1 Pseudo-elementos :before y :after](#)
 - [C.4.22 Sección 12.3.2 Inserción comillas con la propiedad 'content'](#)
 - [C.4.23 Sección 12.6.2 Listas](#)
 - [C.4.24 Sección 14.22 El fondo](#)
 - [C.4.25 Sección 14.2.1 Propiedades del fondo](#)
 - [C.4.26 Sección 16.1](#)
 - [C.4.27 Sección 16.2 Alineación: la propiedad 'text-align'](#)
 - [C.4.28 sección 17.5.1 Capas y transparencia de la tabla](#)
 - [C.4.29 Sección 17.5.2 Algoritmos de la anchura de la tabla](#)

- [C.4.30 Sección 17.6.1 El modelo de bordes separados](#)
- [C.4.31 Fronteras alrededor de las celdas vacías: propiedad 'empty-cells'](#)
- [C.4.32 Sección 17.6.2 El modelo de bordes cerrados](#)
- [C.4.33 Sección 18.2](#)
- [C.4.34 sección A.3](#)
- [C.4.35 Apéndice G.2 Escáner léxico](#)
- [C.4.36 Apéndice E. Referencias](#)
- [**Apéndice D. Hoja de estilo por defecto para HTML 4.0**](#)
- [**Apéndice E. Descripción de elaboración de contextos apilados**](#)
 - [E.1 Definiciones](#)
 - [E.2 Orden de la pintura](#)
 - [E.3 Notas](#)
- [**Apéndice F. Tabla completa de las propiedades**](#)
- [**Apéndice G. Gramática de CSS2.1**](#)
 - [G.1 Gramática](#)
 - [G.2 Escaner léxico](#)
 - [G.3 Comparación de comandos en CSS 2.1 y CSS1](#)
- [**Apéndice I. Index**](#)

1 Acerca de la especificación CSS 2.1

Contenido

- [1.1 CSS 2.1 contra CSS 2](#)
- [1.2 Lectura de la especificación](#)
- [1.3 Cómo se organiza la especificación](#)
- [1.4 Convenciones](#)
 - [1.4.1 Elementos y atributos en el lenguaje del documento](#)
 - [1.4.2 Definiciones de las propiedades de CSS](#)
 - [Valor](#)
 - [Inicial](#)
 - [Se aplica a](#)
 - [Se hereda](#)
 - [Porcentaje](#)
 - [Grupos de medios](#)
 - [Valor computado](#)
 - [1.4.3 Propiedades resumidas](#)
 - [1.4.4 Notas y ejemplos](#)
 - [1.4.5 Imágenes y descripciones largas](#)
- [1.5 Reconocimientos](#)
- [1.6 Aviso De Copyright](#)

1.1 CSS 2.1 contra CSS 2

La comunidad de CSS ha ganado experiencia significativa con la especificación CSS2 desde que se convirtió en una recomendación en 1998. Los errores en la especificación CSS2 se han corregido posteriormente mediante la publicación de varias erratas, pero todavía no ha habido una oportunidad para que la especificación sea cambiada en base a la experiencia ganada.

Mientras que muchas de estas correcciones serán tratadas por la próxima especificación CSS3, la situación actual obstaculiza la implementación (puesta en práctica) y la interoperabilidad de CSS2. Las tentativas de la especificación CSS 2.1 de tratar esta situación es por:

- Compatibilidad al mantener especificaciones de CSS2 que son aceptada e implementadas extensamente.
- Incorporar todas las erratas publicadas de CSS2.
- Indicar donde las implementaciones se diferencia de forma aplastante de la especificación CSS2, modificando la especificación para estar de acuerdo con la práctica generalmente aceptada.
- Quitar las propiedades CSS2 que, en virtud de no ser implementadas, han sido rechazadas por la comunidad de CSS. CSS2.1 apunta a reflejar que propiedades de CSS son razonablemente implementadas para los lenguajes HTML y XML en general (más bien para un lenguaje particular de XML, o *solamente* para el HTML).
- Eliminar las propiedades de CSS2 que serán anticuadas por CSS3, así animando la adopción del CSS3 propuesto en su lugar.
- Añadiendo un número (muy) pequeño [de los nuevos valores de propiedad](#), cuando la experiencia de la puesta en práctica ha demostrado que son necesarias para CSS2.

Así, que si se sigue trabajando con CSS2 se puede perder compatibilidad en un futuro, por ello es recomendable utilizar CSS2.1 que nos permite preservar compatibilidad en el futuro. Mientras que la rotura de compatibilidad no es deseable, creemos que los avances de la revisión en CSS2.1 valen la pena.

1.2 Lectura de la especificación

Esta especificación se ha escrito con dos tipos de lectores en mente: Autores de CSS e implementadores del CSS. Esperamos que la especificación provea a los autores de las herramientas necesarias para componer documentos eficaces, atractivos, y accesibles, sin agobiarlos con detalles de la implementación de CSS. Los implementadores, sin embargo, deben encontrar todo lo necesario para desarrollar [aplicaciones de usuario compatibles](#). La especificación comienza con una presentación general de CSS y se va haciendo más y más técnica y específica. Para el acceso rápido a la información, una tabla general de contenidos, tablas específicas de contenido al principio de cada sección, y un índice proporcionan la navegación fácil, en las versiones electrónicas e impresas.

La especificación se ha escrito con dos modos de presentación en mente: electrónico e impreso. Aunque las dos presentaciones sin duda, serán similares, los lectores encontrarán algunas diferencias. Por ejemplo, los vínculos no funcionan en la versión impresa (obviamente), y los números de páginas no aparecerán en la versión electrónica. En caso de discrepancia, la versión electrónica se considera la versión oficial del documento.

1.3 Cómo se organiza la especificación

La especificación se organiza en las secciones siguientes:

Sección 2: Una introducción a CSS2.1

La introducción incluye un breve tutorial sobre CSS2.1 y una discusión de los fundamentos del diseño que se encuentra detrás de CSS2.1.

Secciones 3 - 20: Manual de referencia de CSS 2.1.

El cuerpo del manual de referencia consiste en la referencia del lenguaje CSS 2.1. Esta referencia define qué puede incluirse en una hoja de estilo CSS 2.1 (sintaxis, propiedades, valores de propiedades) y cómo las aplicaciones del usuario deben interpretar estas hojas de estilo para aspirar a su [conformidad](#).

Apéndices:

Los apéndices contienen la información sobre [propiedades sonoras](#) (no-normativas), [una hoja de estilo de ejemplo para HTML 4.0](#), [cambios de CSS2](#), [la gramática de CSS 2.1](#), una lista de [referencias](#) normativas e informativas, y dos índices: uno para [las propiedades](#) y un [índice general](#).

1.4 Convenciones

1.4.1 Elementos y atributos en el [lenguaje del documento](#)

- Las propiedades CSS, los descriptores, y los nombres de pseudo-clase son delimitados por comillas simples (apóstrofes).
- Los valores de CSS son delimitados por comillas simples (apóstrofes).
- Los nombres de los elementos del lenguaje del documento están en letras mayúsculas.

- Los nombres de los atributos del lenguaje del documento están en letras minúsculas y son delimitados por comillas dobles.

1.4.2 Definiciones de las propiedades de CSS

Cada definición de una propiedad de CSS comienza con un resumen de la información clave dominante que se asemeja a la siguiente:

' nombre_de_la_propiedad '

<i>Valor:</i>	valores permitidos y sintaxis
<i>Inicial:</i>	valor inicial
<i>Se aplica a:</i>	elementos a los cuales se aplica esta propiedad
<i>Se Hereda:</i>	si la propiedad es heredada
<i>Porcentajes:</i>	cómo se interpretan los valores de porcentaje
<i>Medios:</i>	A qué grupo de medios se aplica la propiedad
<i>Valor computado:</i>	cómo computar el valor computado

Valor

Esta parte especifica el conjunto de los valores válidos para la propiedad cuyo nombre es 'nombre_de_la_propiedad'. Los tipos de valor se pueden designar de varias maneras:

1. valores de palabras clave (ej., auto, disco, etc.)
2. tipos de datos básicos, que aparecen entre "<" y ">" (Ej., <medida>, <porcentaje>, etc.). En la versión electrónica del documento, cada instancia de un dato básico contiene un vínculo a su definición.
3. tipos que tienen la misma gama de valores que una propiedad que lleva el mismo nombre (Ej., <'border-width' > <'background-attachment'>, etc.). En este caso, el nombre del tipo es el nombre de la propiedad (completo con comillas) entre "<" y ">" (Ej., <'border-width'>). Dicho tipo no incluye el valor 'heredado'. En la versión electrónica del documento, cada caso de este tipo no-terminal es un vínculo a la correspondiente definición de la propiedad
4. no-terminales que no comparten el mismo nombre que una propiedad. En este caso, el nombre no-terminal aparece entre "<" y ">", como en <border-width>. Note la distinción entre la <border-width> y <'border-width'>; el último se define en los términos del anterior. La definición de un no-terminal se encuentra en su primera aparición dentro de la especificación. En la versión electrónica del documento, cada instancia de este tipo de valor es un vínculo a la correspondiente definición del valor.

Otras palabras en estas definiciones son las palabras claves que deben aparecer literalmente, sin las comillas (Ej., red). La raya vertical (/) y la coma (,) deben también aparecer literalmente.

Los valores pueden estar organizados de las siguientes maneras:

- Varias palabras yuxtapuestas significa que todas ellas deben aparecer, en la orden dado.
- Una barra (|) separa dos o más alternativas: exactamente una de ellas debe aparecer.
- Una barra del doble (||) separa dos o más opciones: una o más de ellas debe aparecer, en cualquier orden.
- Los corchetes ([]) están para agrupar.

La yuxtaposición es más fuerte que la barra doble, y la barra doble es más fuerte que la barra. Así, las líneas siguientes son equivalentes:

$$\begin{array}{c} a\ b \quad | \quad c \quad || \quad d\ e \\ [a\ b] \quad | \quad [c \quad || \quad [d\ e]] \end{array}$$

Cada tipo, palabra clave, o grupo encerrado entre corchetes puede ser seguido por uno de los modificantes siguientes:

- Un asterisco (*) indica que el tipo, la palabra, o el grupo precedente aparece cero o más vez.
- Un signo más (+) indica que el tipo, la palabra, o el grupo precedente aparece una o más veces.
- Un signo de interrogación (?) indica que el tipo, la palabra, o el grupo precedente es opcional.
- Un par de números entre llaves ({ A,B }) indica que el tipo, la palabra, o el grupo precedente aparece por lo menos A y a lo sumo B veces.

Los ejemplos siguientes ilustran diversos tipos del valor:

Valor: N | NW | NE

Valor: [<medida> | thick | thin]{1,4}

Valor: [<family-name>,] * <family-name>

Valor: <uri> ? <color> [/ <color>]

Valor: <uri> || <color>

Los tipos del valor se especifican en términos del símbolo, según lo descrito en [el apéndice G.2](#). Como la gramática permite espacios entre el símbolo en los componentes de la `expr` producción, los espacios pueden aparecer entre los símbolos en valores.

Nota: En muchos casos, los espacios entre símbolos pueden ser necesarios para distinguir los unos de los otros. Por ejemplo, el valor '1em2em' sería analizado como un solo `DIMEN` símbolo con el número '1' y el identificador 'em2em', que sería una unidad no válida. En este caso, sería necesario un espacio antes del '2' para conseguir que lo tome como dos medidas '1em' y '2em'.

Inicial

Esta parte especifica el valor inicial de la propiedad. Si la propiedad se hereda, éste es el valor que se da al elemento raíz [de la estructura del documento](#). Consulte por favor la sección sobre el concepto [cascada](#) para obtener información sobre la interacción entre valores específicos en las hojas de estilo, valores heredados y valores iniciales.

Se aplica a

Esta parte registra los elementos a los cuales se aplica la propiedad. Todos los elementos son considerados aptos para tener todas las propiedades, pero algunas propiedades no tienen ningún efecto sobre el procesamiento de algunos tipos de elementos. Por ejemplo, la propiedad `'clear'` sólo afecta a los elementos a nivel de bloque.

Se hereda

Esta parte indica si el valor de la propiedad está heredado de un elemento antepasado. Consulte por favor la sección sobre el concepto de [cascada](#) para obtener información sobre la interacción entre valores especificados en las hojas de estilo, valores heredados y valores iniciales.

Porcentaje

Esta parte indica cómo deben ser interpretados los porcentajes cuando aparecen en el valor de la propiedad. Si aquí aparece "N/A", significa que la propiedad no acepta porcentajes como valores.

Grupos de medios

Esta parte indica los [grupos de medios](#) a los cuales se aplica la propiedad. La información sobre grupos de medios es no-normativa.

Valor computado

Esta parte describe el valor computado para la propiedad. Vea la sección sobre [los valores computados](#) para ver cómo se utiliza esta definición.

1.4.3 Propiedades resumidas

Algunas propiedades son fórmulas resumidas, significando que permiten a los autores especificar los valores de varias propiedades con una sola propiedad.

Por ejemplo, la propiedad 'font' es una propiedad resumida para definir 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' y 'font-family', todas de una sola vez.

Cuando algunos valores son omitidos en la fórmula resumida, a cada propiedad "ausente" le es asignado su valor inicial (ver la sección sobre [cascada](#)).

Las múltiples reglas de estilo de este ejemplo:

```
H1 {
  font-weight: bold;
  font-size: 12pt;
  line-height: 14pt;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal;
}
```

puede ser reescrito con una sola propiedad resumida:

```
H1 { font: bold 12pt/14pt Helvetica }
```

En este ejemplo, 'font-variant' y 'font-style' toman sus valores iniciales.

1.4.4 Notas y ejemplos

Todos los ejemplos que ilustran usos no permitidos están claramente señalados como "EJEMPLO ILICITO".

Todos los ejemplos de HTML están ajustados a la especificación HTML 4.0 strict DTD (definida en [HTML40](#)) a menos que se indique por la declaración del tipo de documento.

Los ejemplos y notas marcados dentro del código HTML para la especificación y CSS1 las aplicaciones del usuario CSS1 serán procesados por las aplicaciones de usuario de modo especial..

1.4.5 Imágenes y descripciones largas

La mayoría de las imágenes en la versión electrónica de esta especificación están acompañadas por "descripciones extensas" de lo que representan. Un vínculo a la descripción extensa es señalado por una "[D]" después de la imagen.

Las imágenes y las descripciones extensas sólo son informativas.

1.5 Reconocimientos

CSS2.1 está basado en CSS2. Ver la [sección de reconocimientos de CSS2](#) para conocer la gente que contribuyó a CSS2.

Nos gustaría dar las gracias a las siguientes personas quienes, con sus aportaciones y retroalimentación en la lista de mail de www-style, nos han ayudado con la creación de esta especificación: Andrew Clover, Bernd Mielke, C. Bottelier, Christian Roth, Christoph Päper, Claus Färber, Coises, Craig Salla, Darren Ferguson, Dylan Schiemann, Etan Wexler, George Lund, James Craig, Jan Eirik Olufsen, Jan Roland Eriksson, Joris Huizer, Joshua Prowse, Kai Lahmann, Kevin Smith, Lachlan Cannon, Lars Knoll, Lauri Raittila, Mark Gallagher, Michael Day, Peter Sheerin, Rijk van Geijtenbeek, Robin Berjon, Scott Montgomery, Shelby Moore, Stuart Ballard, Tom Gilder, Vadim Plessky, and the Open eBook Publication Structure Working Group Editors. Dar también las gracias a Glenn Adams and Susan Lesch, quienes ayudaron a corregir este documento.

Además, nos gustaría dar gracias especiales a fantasai, Ada Chan, and Boris Zbarsky, quienes han dedicado tiempo a CSS2.1

1.6 Aviso de Copyright

[Copyright](#) © 1997-2003 [W3C](#)® ([MIT](#), [ERCIM](#), [Keio](#)), Todos los derechos reservados. [W3C liability](#), [trademark](#), [document use](#) y [software licensing](#) rules apply.

Los documentos del sitio del W3C son proporcionados por los propietarios del copyright bajo la siguiente licencia. Obteniendo, usando y/o copiando este documento, o el documento de W3C del que esta declaración está vinculada, usted acuerda que ha leído, entendido y cumplirá con los siguiente términos y condiciones:

El permiso para usar, copiar y distribuir el contenido de este documento, o del documento del W3C del que esta declaración está vinculada, por cualquier medio, para cualquier propósito y sin cargo o retribución es concedido por este medio, con la condición de incluir lo siguiente en TODAS las copias del documento, o cualquiera de sus partes, que se usen:

1. Un vínculo o URI al documento original del W3C.
2. El anterior aviso de copyright del autor original, si no existe, un aviso con esta fórmula: "Copyright © [\$date-of-document] [World Wide Web Consortium](#), ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)). Todos los derechos reservados." (Hypertext is preferred, but a textual representation is permitted.)
3. *Si existe, el ESTADO del documento del W3C.*

Cuando el espacio lo permita, se debe proporcionar el texto completo de este **AVISO**. Además, se deberá otorgar crédito a los propietarios del copyright por cualquier programa, documento u otros artículos o productos que se elaboren de conformidad con la implementación del contenido de este documento o cualquier parte del mismo.

No se otorga ningún derecho para crear modificaciones o derivados de acuerdo con esta licencia. Sin embargo, si son satisfecho requerimientos adicionales (documentado en [Copyright FAQ](#)), el derecho a crear modificaciones o derivados es a veces concedido por W3C a individuos que cumplen con esas exigencias.

ESTE DOCUMENTO SE ENTREGA "TAL CUAL ESTÁ" Y LOS PROPIETARIOS DEL COPYRIGHT NO OTORGAN REPRESENTACIONES O GARANTÍAS, EXPRESAS O IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A, GARANTÍAS DE COMERCIALIZACIÓN, DE IDONEIDAD PARA DETERMINADO PROPÓSITO, DE NO CONTRAVENCIÓN O TÍTULO; QUE EL CONTENIDO DEL DOCUMENTO ES ADECUADO A CUALQUIER PROPÓSITO; NI QUE LA IMPLEMENTACIÓN DE TALES CONTENIDOS NO INFRINGEN NINGUNA PATENTE DE TERCEROS, PROPIEDAD INTELECTUAL, MARCAS REGISTRADAS U OTROS DERECHOS.

LOS PROPIETARIOS DEL COPYRIGHT NO SERÁN RESPONSABLES POR NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL O DERIVADO DE CUALQUIER USO DE ESTE DOCUMENTO O EL FUNCIONAMIENTO O IMPLEMENTACION DEL CONTENIDO DEL MISMO.

El nombre y la marca registrada de los propietarios del copyright NO pueden ser usados en publicidades o anuncios relacionados a este documento o su contenido sin una autorización previa explícita y por escrito. El derecho al copyright de este documento sigue siendo en todo momento de los propietarios del copyright.

2 Introducción a CSS 2.1

Contenido

- [2.1 Breve guía de CSS 2.1 para HTML](#)
- [2.2 Breve guía de CSS 2.1 para XML](#)
- [2.3 El modelo de proceso del CSS 2.1](#)
 - [2.3.1 El lienzo](#)
 - [2.3.2 Modelo de destinatarios de CSS 2.1](#)
- [2.4 Principios del diseño del CSS](#)

2.1 Breve guía de CSS 2.1 para HTML

En esta guía, le mostraremos cuán fácil puede ser diseñar unas simples hojas de estilo. Para esta guía, necesitará saber un poco del código HTML (ver [\[HTML40\]](#)) y alguna terminología básica de composición gráfica.

Comenzamos con un pequeño documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

Para establecer el color rojo para el texto de los elementos H1, puede escribir las reglas siguientes del CSS:

```
H1 { color: red }
```

Una regla de CSS consiste de dos partes principales: [selector](#) (' H1 ') y declaración ('color: red'). En HTML, los nombres de los elemento son case-insensitive (no distingue entre mayúsculas y minúsculas), así que sería igual trabajar con 'h1' que con 'H1'. La declaración tiene dos partes: propiedad ('color') y valor ('red'). Con tal que el ejemplo anterior intente influir en una sola propiedad necesaria para el procesamiento de un documento HTML, ya lo califica como una hoja de estilo en sí mismo. Combinado con otras hojas de estilo (un rasgo fundamental de CSS es que las hojas de estilo se combinan) determinará la presentación final del documento. Combinado con otras hojas de estilo (una propiedad fundamental de CSS es que las hojas de estilo están combinadas) determinará la presentación final del documento.

La especificación HTML 4.0 define de qué manera las reglas de las hojas de estilo pueden especificarse para los documentos HTML: ya sea dentro del documento HTML o a través de una hoja de estilo externa. Para poner la hoja de estilo dentro del documento, use el elemento STYLE:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <STYLE type="text/css">
      H1 { color: red }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>

```

Para una mayor flexibilidad, recomendamos que los autores especifiquen hojas de estilo externas; estas pueden cambiarse sin modificar el documento fuente HTML y pueden compartirse entre varios documentos. Para vincular una hoja de estilo externa puede usar el elemento LINK:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <LINK rel="stylesheet" href="bach.css" type="text/css"> </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>

```

El elemento LINK especifica:

- el tipo de vinculo: a una hoja de estilo ("stylesheet").
- la localización de la hoja de estilo a través del atributo "href".
- el tipo de hoja de estilo que se vincula: "text/css".

Para mostrar la íntima relación entre una hoja de estilo y el sistema estructurado de etiquetas (marcas), continuaremos usando el elemento STYLE en esta guía. Agreguemos más colores:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <STYLE type="text/css">
      BODY { color: black; background: white }
      H1 { color: red; background: white }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>

```

La hoja del estilo ahora contiene cuatro reglas: las dos primeras establecen el color y el fondo del elemento BODY como 'black' (negro) y 'white' (blanco) respectivamente (es una buena idea fijar el color del texto y el color del fondo juntos), mientras que los dos siguientes establecen el color y el fondo del elemento H1, como 'red' (rojo) y 'white' (blanco). Puesto que no se ha especificado ningún color para el elemento P, este (P) heredará el color de su elemento padre, BODY. El elemento H1 es también un elemento hijo de BODY pero la segunda regla elimina el valor heredado. En el CSS existen a menudo tales conflictos entre diferentes valores y esta especificación describe cómo resolverlos.

CSS 2.1 tiene más de 90 propiedades diferentes, incluyendo 'color'. Veamos algunos otros:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Bach's home page</TITLE>
    <STYLE type="text/css">
      body {font-family: "Gill Sans", sans-serif;
           font-size: 12pt;
           margin: 3em;
          }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>Bach's home page</H1>
    <P>Johann Sebastian Bach was a prolific composer. </BODY>
  </HTML>
```

Lo primero que se puede observar es que varias declaraciones se agrupan en un bloque encerrado por llaves ({...}) y separado por un punto y coma, no obstante la última declaración también debe estar seguida por un punto y coma.

La primera declaración en el elemento BODY establece la familia de fuentes "Gill Sans". Si esa fuente no está disponible, la aplicación del usuario (a menudo llamada "navegador" ("browser")) usará la familia 'sans-serif' que es una de las cinco familias de fuentes genéricas que todas las aplicaciones del usuario reconocen. Los elementos hijos de BODY heredarán el valor de la propiedad 'font-family'.

La segunda declaración establece el tamaño de la fuente del elemento BODY en 12 puntos. La unidad de medida "punto" es normalmente usada en la tipografía impresa para indicar el tamaño de la fuente y otras dimensiones. Es un ejemplo de unidad de medida absoluta que no varía según el entorno.

La tercera declaración usa una unidad de medida relativa que varía de acuerdo a las circunstancias. La unidad "em" está relacionada con el tamaño de la fuente del elemento. En este caso el resultado es que el margen alrededor del elemento BODY es tres veces mayor que el tamaño de la fuente.

2.2 Breve guía de CSS 2.1 para XML

CSS puede usarse con cualquier formato de documento estructurado, por ejemplo con aplicaciones de eXtensible Markup Language [XML10]. De hecho, XML depende más de las hojas de estilo que HTML puesto que los autores pueden construir elementos propios que las aplicaciones del usuario no saben como mostrar.

Aquí está un fragmento simple de XML:

```
<ARTICULO>
<TITULO>Federico el Grande se encuentra con Bach</TITULO>
<AUTOR>Johann Nikolaus Forkel</AUTOR>
<PARRAFO>
  Una tarde, justo cuando preparaba su <INSTRUMENTO>flauta</INSTRUMENTO>
  y sus músicos estaban dispuestos, un oficial le alcanzó una lista de los
  forasteros que habían arribado.
</PARRAFO>
</ARTICULO>
```

Para que este fragmento sea visualizado correctamente en un documento, debemos declarar primero cuáles son los elementos (inline-level) a nivel de línea (es decir, que no dan lugar a saltos de líneas) y cuáles son (block-level) a nivel de bloque (es decir, que producen saltos de línea).

```
INSTRUMENTO { display: inline }
ARTICULO, TITULO, AUTOR, PARRAFO { display: block }
```

La primera regla declara que INSTRUMENTO será a nivel de línea y la segunda regla, con su lista de selectores separados por comas, declara que todos los demás elementos serán a nivel de bloque.

Nota del traductor:

Los elementos de **bloque** generan una nueva línea (renglón) en el flujo del texto. Dependiendo del tipo de elemento, el navegador puede insertar automáticamente una distancia mayor al contenido antepuesto. Por ejemplo los navegadores generan con el elemento `p` un espacio visible, para que el cambio de párrafo se distinga de un intervalo común. Con el elemento `div` se comienza al contrario solamente una nueva línea.

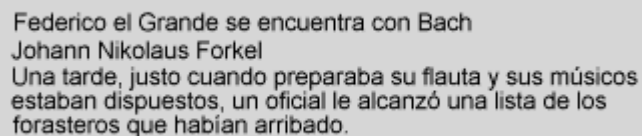
Los elementos de bloque pueden contener generalmente texto normal y **elementos inline**. Algunos elementos de bloque pueden contener también otros elementos de bloque. Excepciones van a ser tratadas en la descripción de los elementos particulares.

Los elementos **inline** no generan una nueva línea en el flujo del texto. Los elementos inline son vistos como elementos "internos" subordinados para los **elementos de bloque**. Generalmente ellos pueden contener en la mayoría de los casos texto normal y otros elementos inline, sin embargo ningún elemento de bloque. Excepciones y restricciones serán tratadas en las respectivas descripciones de los elementos.

Una de las ideas de vincular una hoja de estilo a un documento XML es emplear una instrucción de procesamiento:

```
<?XML:stylesheet type="text/css" href="bach.css"?>
<ARTICULO>
  <TITULO>Federico el Grande se encuentra con Bach</TITULO>
  <AUTOR>Johann Nikolaus Forkel</AUTOR>
  <PARRAFO>Una tarde, justo cuando preparaba su
<INSTRUMENTO>flauta</INSTRUMENTO> y sus músicos estaban dispuestos, un
oficial le alcanzó una lista de los forasteros que habían arribado.
  </PARRAFO>
</ARTICULO>
```

Una aplicación del usuario visual dará este formato al ejemplo anterior:



Federico el Grande se encuentra con Bach
Johann Nikolaus Forkel
Una tarde, justo cuando preparaba su flauta y sus músicos
estaban dispuestos, un oficial le alcanzó una lista de los
forasteros que habían arribado.

[D]

Observe que la palabra "flauta" permanece dentro del párrafo debido a que es el contenido del elemento a nivel de línea INSTRUMENTO.

Pero aún el texto no tiene el formato que se podría esperar. Por ejemplo, el tamaño de la fuente del título debería ser mayor que la del resto del texto y quizás se prefiera mostrar el nombre del autor en itálica:

```
INSTRUMENTO { display: inline }
ARTICULO, TITULO, AUTOR, PARRAFO { display: block }
TITULO { font-size: 1.3em }
AUTOR { font-style: italic }
ARTICULO, TITULO, AUTOR, PARRAFO { margin: 0.5em }
```

Una aplicación del usuario visual dará este formato al ejemplo anterior:



Federico el Grande se encuentra con Bach

Johann Nikolaus Forkel

Una tarde, justo cuando preparaba su flauta y sus músicos
estaban dispuestos, un oficial le alcanzó una lista de los
forasteros que habían arribado.

Agregando más reglas a la hoja de estilo le permitirá mejorar aún más la presentación del documento.

2.3 El modelo de procesamiento de CSS 2.1

Esta sección presenta un posible modelo de cómo trabajan las aplicaciones del usuario que soportan CSS. Este es sólo un modelo conceptual; las implementaciones reales pueden variar.

En este modelo, una aplicación del usuario procesa una fuente atravesando los siguientes pasos:

1. Analizar el documento fuente y crea una [estructura del documento](#) .
2. Identificar el tipo [de medios](#) al que está dirigido.
3. Recuperar todas las hojas de estilo asociadas al documento que se especifican para el tipo [de medios](#) al que está dirigido.
4. Anotar cada elemento de la estructura del documento asignando un solo valor a cada [propiedad](#) que sea aplicable al tipo [de medio](#) al que está dirigido. Los valores de las propiedades se asignan de acuerdo a los mecanismos descritos en la sección de [cascada y herencia](#) .

Parte del cálculo de los valores depende del algoritmo del formato apropiado para el tipo [de medios](#) al que está dirigido. Por ejemplo, si el medio al que se dirige es la pantalla, las aplicaciones del usuario aplican [el modelo de formato visual](#) .

5. De la estructura del documento registrada, genera una estructura de formato. A menudo, la estructura de formato se parece mucho a la estructura del documento, pero también puede diferir en forma significativa, especialmente cuando los autores hacen uso de pseudo-elementos y contenidos generados. Primero, la estructura de formato no necesita ser en forma de "árbol" --la naturaleza de la estructura depende de la implementación--. Segundo, la estructura de formato puede contener mayor o menor información que la estructura del documento. Por ejemplo, si un elemento en la estructura del documento tiene un valor de 'none' (nada) para la propiedad 'display', ese elemento no generará nada en la estructura de formato. Un elemento de lista, por otro lado, puede generar más información en la estructura de formato: el contenido del elemento y la información del estilo de lista (ej., la imagen de la viñeta).

Observe que las aplicaciones del usuario CSS no alteran la estructura del documento durante esta fase. En particular, el contenido generado por causa de las hojas de estilo no es enviado nuevamente al procesador del lenguaje del documento (ej., para un nuevo análisis).

6. Transferir la estructura de formato al medio al que está dirigido (Ej., imprima los resultados, lo muestra en pantalla, lo procesa como palabras, etc.).

El paso 1 fuera del alcance de esta especificación (véase, por ejemplo, [\[DOM 1\]](#)).

Los pasos 2-5 son tratados por el cuerpo de esta especificación.

El paso 6 está fuera del alcance de esta especificación.

2.3.1 El Lienzo

En todos los medios, el término lienzo describe "el espacio donde la estructura del formato es procesada". El lienzo es infinito por cada dimensión del espacio, pero el procesamiento generalmente tiene lugar en una región limitada del lienzo, establecida por la aplicación del usuario de acuerdo con el medio al que está dirigido. Por ejemplo, las aplicaciones del usuario que procesan para pantalla, generalmente imponen un ancho mínimo y eligen un ancho inicial basado en las dimensiones del [acceso visual](#). Las aplicaciones del usuario que procesan una página generalmente imponen restricciones en el alto y el ancho. Las aplicaciones auditivas pueden imponer límites en el terreno del audio, pero no en el tiempo.

2.3.2 Modelo de destinatarios de CSS 2.1

Los [selectores](#) y las propiedades de CSS 2.1 permiten que las hojas de estilo se refieran a las partes siguientes de un documento o aplicación del usuario:

- Los elementos de la estructura del documento y ciertas relaciones entre ellos (ver la sección de [los selectores](#)).
- Los atributos de los elementos de la estructura del documento y los valores de esos atributos (ver la sección de [los selectores de atributos](#)).
- Algunas partes del contenido de un elemento (ver los pseudo-elementos [:first-line](#) y [:first-letter](#)).
- Los elementos de la estructura del documento cuando se encuentran en cierto estado (ver la sección de [pseudo-clases](#)).
- Algunos aspectos del [lienzo](#) en el cual el documento será procesado.
- Alguna información del sistema (ver la sección de [interfaz de usuario](#)).

2.4 Reglas de diseño de CSS

CSS2.1, como CSS2 y como fue antes CSS1, se basa en una serie de reglas de diseño:

- **Compatibilidad hacia atrás y hacia adelante.** Las aplicaciones del usuario CSS2.1 serán capaces de entender las hojas de estilo CSS1. Las aplicaciones del usuario CSS1 podrán leer las hojas de estilo CSS2.1 y descartar las partes que no entienda. Además, las aplicaciones del usuario que no soporten CSS serán capaces de mostrar los documentos estilísticamente mejorados. Por supuesto, los efectos estilísticos hechos posible por CSS no serán procesados, pero todo el contenido será presentado.
- **Complementariedad con documentos estructurados.** Las hojas de estilo complementan los documentos estructurados (ej., HTML y aplicaciones XML) proveyendo información estilística del texto marcado. Debe ser fácil cambiar la hoja de estilo con poco o ningún impacto en el sistema de marcas.
- **Independencia del vendedor, la plataforma y el dispositivo.** Las hojas de estilo permiten a los documentos permanecer independientes del vendedor, la plataforma y el dispositivo. Las mismas hojas de estilo son también independientes del vendedor y la plataforma, pero CSS2.1 permite dirigir una hoja de estilo a un grupo de dispositivos (ej., impresoras).

- **Mantenimiento.** Apuntando a una hoja de estilo desde los documentos, los webmasters (responsables de los sitios Web) pueden simplificar el mantenimiento y conservar un estilo y un efecto consistente a todo lo largo del sitio. Por ejemplo, si el color del fondo de las páginas de una organización cambia, sólo es necesario cambiar un archivo.
- **Simplicidad.** CSS es un lenguaje de estilo simple que es humanamente legible y posible de escribir. Las propiedades CSS se mantienen independientes unas de otras en la medida de lo posible y generalmente sólo hay un modo de conseguir un efecto determinado.
- **Rendimiento de la red.** CSS proporciona una compacta codificación para presentar los contenidos. Comparado con los archivos de imágenes o de audio que son usados frecuentemente por los autores para conseguir ciertos efectos en el procesamiento, las hojas de estilo, la mayoría de las veces, disminuyen el tamaño del contenido. Además, tienen que ser abiertas menos conexiones de red, lo cual ayuda a incrementar el rendimiento de la red.
- **Flexibilidad.** Las CSS pueden ser aplicadas al contenido de varias maneras. La propiedad clave es la capacidad de formar estilos en cascada con la información especificada en la hoja de estilo predefinida (aplicación del usuario), las hojas de estilo del usuario, las hojas de estilo vinculadas, el encabezamiento del documento y en los atributos de los elementos que forman el cuerpo del documento.
- **Riqueza.** Proporcionando a los autores un abundante juego de efectos de procesamiento, aumenta la riqueza de la Web como medio de expresión. Los diseñadores han estado ambicionando la funcionalidad comúnmente encontrada en los programas de edición y de presentaciones gráficas. Algunos de los efectos requeridos entran en conflicto con la independencia del dispositivo, pero CSS2.1 llega muy lejos satisfaciendo las demandas de los diseñadores.
- **Combinación con lenguajes alternativos.** El juego de propiedades de CSS descritos en esta especificación conforman un sólido modelo de aplicación de formatos para presentaciones visuales y auditivas. Este modelo puede ser accedido mediante el lenguaje CSS, pero la combinación con otros lenguajes también es posible. Por ejemplo, un programa en JavaScript puede cambiar dinámicamente el valor de la propiedad 'color' de un determinado elemento.
- **Accesibilidad.** Varias propiedades de CSS harán a la Web más accesible para los usuarios con discapacidades:
 - Las propiedades para controlar la apariencia de las fuentes permiten a los autores eliminar las inaccesibles imágenes de texto procesado.
 - Las propiedades de posicionamiento permite a los autores eliminar los artilugios con el sistema de marcas (ej., imágenes invisibles) para forzar la composición.
 - La semántica de las reglas !important indica que los usuarios con particulares requerimientos de presentación pueden suplantar las hojas de estilo del autor.
 - El nuevo valor 'inherit' (heredado) para todas las propiedades mejora la generalidad del funcionamiento en cascada y permite una más fácil y consistente sintonía en el estilo.
 - El avanzado soporte de medios, incluyendo grupos de medios y los tipos de medios braille, de relieve y tty permitirán a los usuarios y autores confeccionar páginas para esos dispositivos.

Note. Para mayor información acerca del diseño de documentos accesibles usando CSS y HTML, por favor, consulte [\[WAI-PAGEAUTH\]](#).

3 Conformidad: Requisitos y recomendaciones

Contenido

- [3.1 Definiciones](#)
- [3.2 Conformidad](#)
- [3.3 Condiciones de error](#)
- [3.4 Tipo de contenido text/css](#)

3.1 Definiciones

En esta sección, comenzamos la especificación formal de CSS 2.1, comenzando con el contrato entre los autores, usuarios, e implementadores.

Las palabras claves "DEBEN", "NO DEBE", "REQUERIDO", "DEBERÍA", "NO DEBERÍA", "DEBERÁ", "NO DEBARÁ", "RECOMENDADÓ", "PUEDE" y "OPCIONAL" en este documento serán interpretadas tal como se escribe en RFC 2119 (véase [\[RFC2119\]](#)). Sin embargo, por razones de legibilidad, estas palabras no aparecen en mayúsculas en esta especificación.

Ocasionalmente, esta especificación recomienda la buena práctica para los autores y las aplicaciones del usuario. Estas recomendaciones no son normativas y la conformidad con esta especificación no depende de su cumplimiento. Estas recomendaciones contienen la expresión "que recomendamos...", "esta especificación recomienda...", o alguna frase similar.

Hoja de estilo

Un conjunto de estamentos que especifican la forma de presentación de un documento.

Las hojas de estilo pueden tener tres orígenes distintos: [autor](#), [usuario](#), y [aplicación del usuario](#). La interacción de estas fuentes se describe en la sección sobre [cascada y herencia](#).

Hoja de estilo valida

La validez de una hoja de estilo depende del nivel del CSS usado para la hoja de estilo. Todas las hojas de estilo CSS1 válidas son hojas de estilo de CSS 2.1 válidas, pero algunos cambios efectuados sobre CSS1 significan que algunas hojas de estilo CSS1 tendrán una semántica ligeramente diferente en CSS 2.1. Algunas propiedades en CSS2 no son parte de CSS 2.1, así que no todas las hojas de estilo CSS2 son hojas válidas de estilo CSS 2.1.

Una hoja de estilo válida en CSS 2.1 se debe escribir según [la gramática de CSS 2.1](#). Además, debe contener solo reglas-arroba, nombres de propiedades, y los valores de propiedades definidos en esta especificación. Una regla-arroba, nombre de propiedad, o valor de propiedad ilícito es aquella que no es válida.

Documento fuente

El documento al cual se aplica una o más hojas de estilo. Este está codificado en algún lenguaje que representa al documento como una estructura de [elementos](#). Cada elemento consta de un nombre que identifica el tipo de elemento, opcionalmente algunos [atributos](#), y [contenido](#) (posiblemente vacío). Por ejemplo, el documento fuente podía ser un caso de XML o del SGML.

Lenguaje del documento

El lenguaje de codificación del documento fuente (Ej., HTML, XHTML o SVG). CSS se emplea para describir la presentación de los lenguajes del documento y CSS no cambia la semántica subyacente de los lenguajes del documento.

Elemento

(un término SGML, ver [\[ISO8879\]](#).) Las estructuras sintácticas primarias del lenguaje del documento. La mayoría de las reglas de las hojas de estilo CSS usan el nombre de estos elementos (como "P", "TABLE" y "OL" para HTML) para especificar la información acerca de su procesamiento.

Elemento sustituido

Un elemento del cual el intérprete de CSS solo conoce las dimensiones intrínsecas. En HTML, los elementos de IMG y OBJETO pueden ser elementos sustituidos. Por ejemplo, el contenido del elemento de IMG es sustituido a menudo por la imagen que el atributo "src" designa.

Dimensiones intrínsecas

La anchura y la altura definidos por el propio elemento, no impuesta por el entorno. CSS no define cómo se encuentran las dimensiones intrínsecas. En CSS 2.1 se asume que todos los elementos sustituidos, y solamente los elementos sustituidos, poseen dimensiones intrínsecas.

Atributo

Un valor asociado a un elemento, consiste en un nombre, y un valor (textual) asociado.

Contenido

El contenido asociado a un elemento en el documento fuente; no todos los elementos tienen contenido, en cuyo caso son llamados vacíos. El contenido de un elemento puede incluir texto y puede incluir varios sub-elementos, en cuyo caso el elemento es llamado padre de esos sub-elementos.

Ignorar

Este término tiene dos significados levemente diferentes. Primero, un programa de análisis de CSS debe seguir ciertas reglas cuando descubre sintaxis desconocidas o ilícitas en una hoja de estilo. El programa de análisis debe entonces no hacer caso de ciertas partes de las hojas de estilo. Las reglas exactas para saber que partes del código no deben ser interpretadas están en la [Las declaraciones y las propiedades, reglas para manejar los errores del análisis, valores no soportados](#), o se pueden explicar en el texto donde aparece el término "no hace caso". En segundo lugar, una aplicación del usuario puede (y, en algunos casos debe) desatender ciertas propiedades o los valores incluso si el sintaxis es legal. Por ejemplo, los elementos del table-column-group no pueden tener bordes alrededor de ellos, así que las propiedades del borde deben ser ignoradas.

Contenido procesado

El contenido de un elemento después del procesamiento que se efectúa sobre él conforme a las hojas de estilo relevantes que han sido aplicadas. El contenido procesado de un [elemento sustituido](#) proviene de fuera del documento fuente. El contenido procesado también puede ser el texto alternativo de un elemento (ej., el valor del atributo HTML "alt"), y puede incluir objetos insertados implícita o explícitamente por la hoja de estilo, como viñetas, numeradores, etc.

Estructura del documento

La estructura de elementos codificados en el documento fuente. Cada elemento en esta estructura tiene exactamente un padre, con la excepción del elemento **raíz** que no tiene ninguno.

Hijo

Un elemento A es llamado hijo de un elemento B si, y sólo si, B es el padre de A.

Descendiente

Un elemento A se llama descendiente de un elemento B, si (1) A es un hijo de B, (2) A es el hijo de un cierto elemento C que sea un descendiente de B.

Antepasado

Un elemento A se llama antepasado de un elemento B, si y solo si B es un descendiente de A.

Hermano

Un elemento A es llamado hermano de un elemento B si, y sólo si, B y A comparten el mismo elemento padre. Un elemento A es hermano precedente si viene antes que B en la estructura del documento. Un elemento A es hermano siguiente si viene después de B en la estructura del documento.

Elemento precedente

Un elemento A es llamado elemento precedente de un elemento B si, y sólo si, (1) A es un antepasado (antecesor) de B o (2) A es hermano precedente de B.

Elemento siguiente

Un elemento A es llamado elemento siguiente de un elemento B si, y sólo si, B es un elemento precedente de A.

Autor

Un autor es una persona que escribe documentos y las hojas de estilo asociadas. Una herramienta de autor genera documentos y las hojas de estilo asociadas.

Usuario

Un usuario es una persona que interactúa con una aplicación de usuario para ver, oír o usar de algún modo un documento y sus hojas de estilo asociadas. El usuario puede proporcionar una hoja de estilo personal que codifica sus preferencias individuales.

Aplicación del usuario (UA)

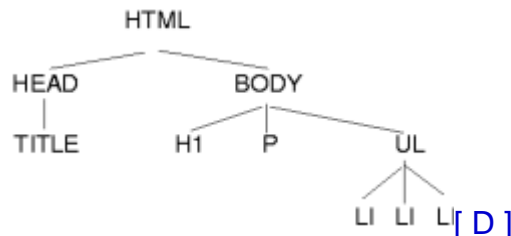
Una aplicación del usuario es cualquier programa que interprete un documento escrito en la lenguaje del documento y aplica las hojas de estilo asociadas según los términos de esta especificación. Una aplicación del usuario puede mostrar un documento, leerlo en voz alta, permitir que sea impreso, convertirlo a otro formato, etc.

Una aplicación de usuario en HTML es uno que apoya las especificaciones de HTML 2.x, de HTML 3.x, o de HTML 4.x. Una aplicación de usuario que apoya XHTML [[XHTML](#)], pero no el HTML (según lo enumerado en la oración anterior) no se considera una aplicación de usuario de HTML de conformidad con esta especificación.

Aquí está un ejemplo de un documento fuente escrito en el HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <TITLE>Mi página personal</TITLE>
  <BODY>
    <H1>Mi página personal</H1>
    <P>¡Bienvenidos a mi página! Permítanme hablarles de mis compositores
favoritos:
  <UL>
    <LI> Elvis Costello
    <LI> Johannes Brahms
    <LI> Georges Brassens
  </UL>
</BODY>
</HTML>
```

Esto da lugar al árbol siguiente:



Según la definición de HTML 4,0, los elementos HEAD serán deducidos (inferidos) durante el análisis y convertidos en parte de la estructura del documento incluso si las etiquetas "HEAD" no están en la fuente del documento. De modo similar, el programa analizador (navegador por ejemplo) sabe dónde P y los elementos LI terminan, aunque no haya marcas </p> y en la fuente.

Los documentos escritos en XHTML (y otros lenguajes basados en XML) se comportan de forma diferente: no hay elementos deducidos y todos los elementos deben tener etiquetas del cierre </ etiqueta>.

3.2 Conformidad

Esta sección solo define conformidad (compatibilidad) con la especificación de CSS 2.1. Puede haber otros niveles de CSS que en el futuro requieran a las aplicaciones del usuario implementar un conjunto de propiedades diferentes para obtener conformidad.

En general, los puntos siguientes deben ser observados por las aplicaciones del usuario que pretendan conformidad con esta especificación:

1. Debe soportar uno o más de los [tipos de medios](#) de CSS 2.1 .
2. Para cada documento fuente, debe intentar recuperar todas las hojas de estilo asociadas que sean apropiadas para los tipos de medio soportados. Si no puede recuperar todas las hojas de estilo asociadas (por ejemplo, debido a errores en la red), debe mostrar el documento usando aquellas que puede recuperar.
3. Debe analizar las hojas de estilo de acuerdo con esta especificación. En particular, debe reconocer todas las reglas-arroba, bloques, declaraciones y selectores (ver [la gramática de CSS 2.1](#)). Si una aplicación del usuario encuentra una propiedad que se aplica a un tipo de medio soportado, la aplicación del usuario debe analizar el valor de acuerdo a la definición de la propiedad. Esto significa que la aplicación del usuario debe aceptar todos los valores válidos y debe ignorar las declaraciones con valores no válidos. Las aplicaciones del usuario deben ignorar reglas aplicadas a [tipos de medios](#) que no son soportados.
4. Para cada elemento en una [estructura de documento](#), debe asignar un valor para cada propiedad según las reglas de la propiedad y la definición de [cascada y herencia](#) .
5. Si el documento fuente viene con hojas de estilo alternativas (por ejemplo con la palabra clave "alternate" en HTML 4.0 [[HTML40](#)]), la AU debe permitir que el usuario seleccione una de entre estas hojas de estilo y aplicar la seleccionada.

Sin embargo, no todas las aplicaciones del usuario deben observar cada punto:

- Una aplicación del usuario que lee las hojas de estilo sin la representación de ningún contenido (Ej., un Validador de CSS 2.1) debe respetar los puntos 1-3.
- Una herramienta de autor solo necesita generar [hojas de estilo válidas](#)
- Una aplicación del usuario que procesa un documento con hojas de estilo asociadas debe respetar los puntos 1-5 y procesar el documento según los requisitos específicos de los medios planteados en esta especificación. Los [valores](#) pueden ser aproximados cuando la aplicación del usuario así lo requiera.

La incapacidad de una aplicación del usuario para implementar parte de esta especificación debido a las limitaciones de un dispositivo en particular (ej., una aplicación del usuario no puede presentar colores sobre un monitor monocro o una página) no implica la no conformidad.

Las AUs deben permitir que los usuarios especifiquen un archivo que contenga la hoja de estilo del usuario. Las AUs que funcionan en dispositivos sin ningún medios de escritura o la especificación de archivos son eximidas de estos requisitos. Además, las AUs pueden ofrecer otros medios de especificar preferencias del usuario, por ejemplo a través de un GUI.

CSS2.1 no define que propiedades se aplican a los controles y a los marcos (frames), o cómo CSS se puede utilizar para darles forma. Las aplicaciones del usuario pueden aplicar propiedades del CSS a estos elementos. Los autores recomiendan que esta ayuda se trate como experimental. Un nivel futuro de CSS puede tener otras especificaciones.

3.3 Condiciones de error

En general, este documento no especifica el comportamiento de la gestión de errores para las aplicaciones del usuario (Ej., cómo se comportan cuando no pueden encontrar un recurso señalado por un URI).

Sin embargo, las aplicaciones del usuario deben observar [las reglas para el manejo de los errores de análisis](#).

Puesto que las aplicaciones del usuario pueden variar en el modo de manejar las condiciones de error, los autores y usuarios no deben confiar en un comportamiento específico para la recuperación de un error.

3.4 Tipo de contenido text/css

Las hojas de estilo CSS que existen en archivos separados son enviadas a través de Internet como una secuencia de bytes acompañada por la información de codificación. La estructura de la transmisión, llamada entidad del mensaje, es definida por RFC 2045 y RFC 2068 (ver [RFC2045](#) y [RFC2068](#)). Una entidad del mensaje con un tipo de contenido "text/css" representa un documento CSS independiente. El tipo de contenido "text/css" ha sido registrado por RFC 2318 ([RFC2318](#)).

4 Sintaxis y tipos de datos básicos

Contenido

- [4.1 Sintaxis](#)
 - [4.1.1 Comandos](#)
 - [4.1.2 Palabras claves](#)
 - [Extensiones específicas-navegador](#)
 - [Notas Históricas Informativas](#)
 - [4.1.3 Caracteres y mayúsculas/minúsculas](#)
 - [4.1.4 Declaraciones](#)
 - [4.1.5 Reglas-arroba](#)
 - [4.1.6 Bloques](#)
 - [4.1.7 Sistemas de reglas, bloques de declaraciones, y selectores](#)
 - [4.1.8 Declaraciones y propiedades](#)
 - [4.1.9 Comentarios](#)
- [4.2 Reglas para manejar errores de análisis](#)
- [4.3 Valores](#)
 - [4.3.1 Números enteros y números reales](#)
 - [4.3.2 Medidas](#)
 - [4.3.3 Porcentajes](#)
 - [4.3.4 URL + URN = URI](#)
 - [4.3.5 Contadores](#)
 - [4.3.6 Colores](#)
 - [4.3.7 Cadenas](#)
 - [4.3.8 Valores no soportados](#)
- [4.4 Representación de los documentos CSS](#)
 - [4.4.1 Referencias a los caracteres no representados en una codificación de caracteres](#)

4.1 Sintaxis

Esta sección describe una gramática (y *el análisis con compatibilidad futura*) para cualquier versión de CSS (incluyendo CSS2.1). Las versiones futuras de CSS se adherirán a esta sintaxis base, aunque pueden agregar restricciones sintácticas adicionales.

Estas descripciones son normativas. También son complementadas por las reglas normativas de gramática presentadas en el [apéndice G](#).

4.1.1 Comandos

Todos los niveles de CSS --nivel 1, nivel 2 y cualquier futuro nivel-- usan la misma sintaxis central. Esto permite a las AU analizar (aún cuando no las entiendan completamente) las hojas de estilo escritas en niveles de CSS que no existían en el momento en que la AU fue creada. Los diseñadores pueden usar esta propiedad para crear hojas de estilo que funcionen con aplicaciones de usuario antiguas, mientras ejercitan también las posibilidades de los niveles más nuevos de CSS.

En el nivel léxico, las hojas de estilo CSS consisten en una secuencia de comandos. La lista de comandos para CSS 2.1 es la que sigue. Las definiciones utilizan expresiones regulares al estilo de Lex. Los códigos octales se refieren a ISO 10646 ([\[ISO10646\]](#)). Como en Lex, en caso de múltiples equivalencias, la equivalencia más larga determina el comando.

comando	Definición
IDENT	{ ident }
ATKEYWORD	@ {ident}
STRING	{ string }
HASH	# { name }
NUMBER	{ num }
PERCENTAGE	{ num }%
DIMENSION	{ num } { ident }
URI	url\(\{ w \} { string } { w } \\ url\(\{ W \} ([!#\$%&*~] { nonascii } { escape } * { W })\)
UNICODE-RANGE	U\+[0-9a-f?]{1,6}(-[0-9a-f]{1,6})?
CDO	<!--
CDC	-->
;	;
{	\{
}	\}
(\(
)	\)
[\[
]	\]
S	[\t\r\n\f]+
COMMENT	\/*[^*]**+([^*][^*]**+)*\/
FUNCTION	{ indent } \ (
INCLUDES	~=
DASHMATCH	=
DELIM	cualquier otro carácter no emparejado por las reglas anteriores

Las macros entre llaves ({}) de la tabla se definen como sigue:

Macro	Definición
ident	[-]? { nmstart } { nmchar } *
Name	{ nmchar } +
nmstart	[_ a-zA-Z] { nonascii } { escape }
nonascii	[^\0-\177]
unicode	\\ [0-9a-f] { 1,6 } (\r \n \t [\n \r \t \f]) ?
escape	{ unicode } \ \ [--\200-\4177777]
nmchar	[_ a-zA-Z0-9 -] { nonascii } { escape }
Num	[0-9] + [0-9] * \. [0-9] +
String	{ string1 } { string2 }
string1	\" ([\t ! # \$ % & (~] \\ { nl } \ ' { nonascii } { escape } * \ \"
string2	' ([\t ! # \$ % & (~] \\ { nl } \ \" { nonascii } { escape } * \ '
nl	\n \r \n \r \f
W	[\t \r \n \f] *

Debajo está la sintaxis central para CSS. Las secciones que siguen describen cómo utilizarlo. El [apéndice G](#) describe una gramática más restrictiva que está más cerca al lenguaje de nivel 2 de CSS.

```
Hoja de estilo: [ CDO|CDC|S|declaración ] *;
Declaración   : regla|regla-arroba;
regla-arroba  : ATKEYWORD S* otros * [ bloque|';' S* ];
bloque        : ' { ' S* [ otros|bloque|ATKEYWORD S*|';' S* ]* '}' S*;
regla         : selector? '{' S* declaración? [ ';' S* declaración?
]* '}' S*;
selector      : otros+;
declaración   : cDelim? propiedad S* ':' S* valor;
propiedad     : IDENT;
valor         : [ otros|bloque|ATKEYWORD S* ] +;
otros: [ IDENT|NUMBER|PERCENTAGE|DIMENSION|STRING|DELIM|URI|HASH|UNICODE-
RANGE|INCLUDES|DASHMATCH|FUNCIÓN S * otros * ' ) '| ( ' S * otros * ' ) '|
[ ' S * otros * ' ] ' ] S*;
```

Los comandos COMMENT no aparecen en la gramática (para mantenerla legible), pero cualquier cantidad de estos comandos pueden aparecer en cualquier lugar entre otros comandos.

El comando S en la gramática anterior está colocado para los espacios en blanco. Solamente los caracteres "espacio" (U+0020 -- código 32 en Unicode), "tabulador" (U+0009 -- 9), "avance de línea" (U+000A -- 10), "retorno de carro" (U+000D -- 13) y "avance de hoja" (U+000C -- 12) pueden aparecer en los espacios en blanco. Otros caracteres que producen espacios, tales como "espacio eme" (U+2003 -- 8195) y "espacio ideográfico" (U+3000 -- 12288) nunca forman parte de los espacios en blanco.

4.1.2 Palabras claves

Las palabras claves tienen la forma de identificadores. Las palabras claves no se deben poner entre comillas ("..." o '... '). Así,

```
red
```

es una palabra clave, pero

```
"red"
```

no lo es. (es [una cadena de texto](#).) Otros ejemplos ilícitos:

```
width: "auto";  
border: "none";  
font-family: "serif";  
background: "red";
```

Extensiones Especifica-Navegador

En CSS2.1, los identificadores pueden comenzar con '-' (guión) o '_' (guión bajo). Las palabras claves y nombres de propiedades, que comienzan con '-' o '_' son reservadas para las extensiones específica del navegador. Tales extensiones específicas del navegador deben tener uno de los formatos siguientes:

```
'-' + identificador navegador + '-' + nombre propiedad  
 '_' + identificador navegador + '-' + nombre propiedad
```

Por ejemplo, si la organización XYZ agregó una propiedad, a su navegador, para describir el color del borde derecho de un elemento, esta se puede llamar -xyz-border-east-color .

Otros ejemplos conocidos:

```
-moz-box-sizing  
-moz-border-radius  
-wap-accesskey
```

Un guión (-) o un guión bajo (_) inicial, no siempre es garantía para poder utilizar una propiedad o palabra clave en ningún nivel actual o futuro de las implementaciones de CSS. Así que CSS puede no reconocer tales propiedades y puede no hacer caso de ellas según [las reglas para manejar errores de análisis](#) . Sin embargo, como el guión (-) y el guión bajo (_) inicial es parte de la gramática, los implementadores de CSS2.1 deben siempre poder utilizar un programa de análisis de confirmación de CSS, para comprobar si admite o no cualquier extensión específicas del navegador.

Notas Históricas Informativas

Esta sección es informativa.

A la hora de escribir este manual, los prefijos siguientes son los que se conocen:

prefijo	organización	notas
mso-	Microsoft Corporation	Creado antes del grupo de trabajo estableció la convención de nombramiento para las extensiones.
- moz-	La Organización De Mozilla	
- o-	Software De la Ópera	
- atsc-	Comité De Estándares Avanzado De la Televisión	
- wap-	El Foro de WAP	

Las extensiones específicas de Navegador/Organización deben ser evitadas.

4.1.3 Caracteres y mayúsculas/minúsculas

Las reglas siguientes siempre sostienen:

- Todas las hojas de estilo CSS son insensibles a las mayúsculas/minúsculas (case-insensitives), excepto para las partes que no están bajo el control de CSS. Por ejemplo, la diferenciación entre mayúsculas y minúsculas en los valores de los atributos HTML "id" y "class", los nombres de las fuentes y los URI están fuera del alcance de esta especificación. Adverta en particular que los nombres de los elementos en HTML no hacen distinción entre mayúsculas y minúsculas, pero sí lo hacen en XML.
- En CSS 2.1, los identificadores (incluyendo los nombres de los elementos, clases e ID de los [selectores](#)) pueden contener solamente los caracteres [A-Za-z0-9] y los caracteres U+00A1 (161) en adelante en ISO 10646, más el guión (-) y el guión bajo (_); no pueden comenzar con un número. Solamente las propiedades, los valores, las unidades, las pseudo-clases, los pseudo-elementos, y las reglas-arroba pueden comenzar con un guión (-); otros identificadores (Ej.. nombres, clases, o identificaciones de elemento) no pueden. Los identificadores pueden también contener caracteres escapados y cualquier carácter de la ISO 10646 como código numérico (véase el artículo siguiente). Por ejemplo, el identificador "B&W?" puede ser escrito como "B&W \?" o "B26 W3F".

Observe que Unicode es simétricamente equivalente a ISO 10646 (véase [\[UNICODE\]](#) y [\[ISO10646\]](#)).

- En CSS 2.1, una barra invertida (\) indica tres tipos de escapes de caracteres:

Primero, dentro de una [cadena](#), una barra invertida seguida por una nueva línea es ignorada (es decir, la cadena se evalúa para no contener la barra invertida o la nueva línea).

En segundo lugar, cancela el significado de los caracteres especiales de CSS. Cualquier carácter (excepto un número hexadecimal) se puede escapar con una barra invertida para quitar su significado especial. Por ejemplo, "\ " es una secuencia que consiste en una comilla doble. Los pre-procesadores de las hojas de estilo no deben quitar estas barras invertidas de una hoja de estilo debido a que ello puede cambiar el significado de la misma.

Tercero, el escape con barra invertida permite a los autores referirse a caracteres que no pueden ponerse fácilmente en un documento. En este caso, la barra invertida es seguida hasta por seis números hexadecimales (0..9A..F), que representan los caracteres en ISO 10646 ([\[ISO10646 \]](#)) con ese número, que no debe ser cero. Si un carácter en la gama [0-9a-fA-F] sigue el número hexadecimal, el final del número debe ser declarado. Hay dos maneras de hacer eso:

1. con el espacio (u otro carácter de espacio en blanco): "\26 B" ("&B"). En este caso, las aplicaciones del usuario deben tratar un par de "CR/LF" (U+000D/U+000A) como carácter del espacio en blanco.
2. proporcionando exactamente 6 dígitos hexadecimales: "\000026B" ("&B")

De hecho, estos dos métodos pueden ser combinados. Sólo un carácter de espacio en blanco es ignorado después del escape hexadecimal. Advierta que esto significa que un espacio "real" después de la secuencia de escape debe ir él mismo con escape o duplicado.

- Los escapes con barra invertida se consideran siempre parte de un [identificador](#) o de una cadena (es decir, "\7B" no es puntuación, aun cuando "{" si lo es, y "\32" está permitido al comienzo de un nombre de una clase, aunque "2" no lo está).

4.1.4 Declaraciones

Una hoja de estilo de CSS, para cualquier versión de CSS, consiste en una lista de *declaraciones* (véase la gramática arriba). Hay dos clases de declaraciones: *reglas-arroba* y *sistemas de reglas*. Puede haber [espacios en blanco](#) alrededor de las declaraciones.

En esta especificación, las expresiones "inmediatamente antes" o "inmediatamente después" significan que no intervienen espacios en blanco o comentarios.

4.1.5 Reglas-arroba

Las Reglas-arroba comienzan con *una clave-arroba*, un carácter '@' seguido inmediatamente por un [identificador](#) (por ejemplo, '@import', '@page').

Una regla-arroba consiste en todo que hay hasta, e incluyendo, el punto y coma siguiente (;) o el siguiente [bloque](#), cualquiera que sea el primero en aparecer. Una aplicación del usuario de CSS que encuentra una regla-arroba desconocida debe [ignorar](#) el conjunto de la regla-arroba y continuar el análisis después de ella.

Las aplicaciones de usuario de CSS 2.1 deben [ignorar](#) cualquier regla '@import' que aparezca dentro de un [bloque](#) o que no preceda toda la regla fijada.

Asuma, por ejemplo, que un programa de análisis de CSS 2.1 encuentra esta hoja de estilo:

```
@import "subs.css";
h1 { color: blue;}
@import "list.css";
```

El segundo '@import' es ilícito según CSS2.1. El programa de análisis de CSS 2.1 [ignora](#) la regla-arroba entera, reduciendo con eficacia la hoja de estilo a:

```
@import "subs.css";
h1 { color: blue }
```

En el ejemplo siguiente, la segunda regla '@import' no es válida, puesto que aparece dentro de un bloque de '@media'.

```
@import "subs.css";
@media print {
    @import "print-main.css";
    BODY { font-size: 10pt }
}
H1 {color: blue }
```

4.1.6 Bloques

Un bloque comienza con una llave ({} y termina con la correspondiente llave hacia la derecha (}). En medio puede haber cualquier carácter, excepto que los paréntesis (()), los corchetes ([]) las llaves ({}) deben aparecer siempre con el par correspondiente y pueden anidarse. Las comillas simples (') y dobles (") también deben ir a pares, y los caracteres encerrados por ellas son analizados como una cadena. Ver los [Comandos arriba para la definición de una cadena](#).

Aquí hay un ejemplo de bloque. Observe que la llave derecha entre las comillas dobles no se corresponde con la llave de apertura del bloque, y que la segunda comilla simple es un [carácter escapado](#), y de este modo no se corresponde con la primera comilla simple:

```
{ causta:  }" + ( { 7 } * '\'' ) }
```

Observe que la regla antedicha no es válida en CSS 2.1, pero sigue siendo un bloque según lo definido arriba.

4.1.7 Sistemas de reglas, bloques de declaraciones, y selectores

Un sistema de la reglas (también llamado "regla") consiste en un selector seguido por un bloque de declaraciones.

Un bloque de declaraciones (también llamado bloque-{} en el siguiente texto) comienza con una llave izquierda ({} y termina con la llave derecha (}) correspondiente. En medio de ellas debe haber una lista de cero o más declaraciones separadas por punto y coma (;).

El selector (ver también la sección sobre [selectores](#).) consiste en todo lo que hay hasta (pero sin incluir) la primera llave izquierda ({}). Un selector siempre va junto con un bloque-{}. Cuando una aplicación del usuario no puede analizar el selector (es decir, no es un CSS2.1 válido), debe [ignorar](#) también el bloque-{}.

CSS 2.1 da un significado especial a la coma (,) en los selectores. Sin embargo, puesto que no se sabe si la coma puede adquirir otros significados en las versiones futuras de CSS, la declaración entera debe ser [ignorada](#) si hay un error en cualquier parte del selector, aunque el resto del selector pueda parecer razonable en CSS 2.1.

Por ejemplo, como el "&" no es un comando válido en un selector CSS2.1, la aplicación del usuario CSS2.1 debe [ignorar](#) la segunda línea completa y no poner el color de H3 como rojo:

```
h1, h2 {color: green }
h3, h4 & h5 {color: red }
h6 {color: black }
```

Aquí hay un ejemplo más complejo. Los primeros dos pares de llaves están dentro de una cadena y no señalan el cierre del selector. Este es un estamento CSS2.1 válido.

```
p[example="public class foo\
{\
  private int x;\
\
  foo(int x) {\
    this.x = x;\
  }\
\
}"] { color: red }
```

4.1.8 Declaraciones y propiedades

Una declaración puede ser vacía o consistir en una propiedad, seguida por dos puntos (:), seguidos por un valor. Alrededor de cada uno de estos puede haber [espacio en blanco](#).

Debido a la manera en que trabajan los selectores, las declaraciones múltiples para el mismo selector pueden organizarse en grupos separados por punto y coma (;).

De este modo, las siguientes reglas:

```
h1 { font-weight: bold }
h1 { font-size: 12px }
h1 { line-height: 14px }
h1 { font-family: Helvetica }
h1 { font-variant: normal }
h1 { font-style: normal }
```

son equivalentes a:

```
h1 {
  font-weight: bold;
  font-size: 12px;
  line-height: 14px;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal
}
```

Una propiedad es un [identificador](#). Cualquier carácter puede aparecer en el valor. Los paréntesis ("()"), los corchetes ("[]"), las llaves ("{}"), las comillas simples (') y las comillas dobles (") deben ir con su par correspondiente, y los punto y coma que no formen parte de una cadena deben ser [escapados](#). Los paréntesis, los corchetes y las llaves pueden anidarse. Dentro de las comillas, los caracteres son tomados como una cadena.

La sintaxis de los valores se especifica separadamente para cada propiedad, pero en todos los casos, los valores están compuestos de identificadores, cadenas, números, medidas, porcentajes, URI, colores, ángulos, tiempos y frecuencias.

Una aplicación del usuario debe [ignorar](#) una declaración con un nombre de propiedad no válido o un valor ilícito. Cada propiedad CSS2.1 tiene sus propias restricciones sintácticas y semánticas para los valores que acepta.

Por ejemplo, asumiendo que un analizador CSS2.1 encuentra esta hoja de estilo:

```
h1 { color: red; font-style: 12pt } /*valor no valido: 12pt */
p { color: blue; font-vendor: any; /*propiedad no valida: font-vendor */
  font-variant: small-caps }
em em { font-style: normal }
```

La segunda declaración en la primera línea contiene un valor ilícito: '12pt'. La segunda declaración en la segunda línea contiene una propiedad indefinida 'font-vendor'. El analizador CSS2.1 [ignorará](#) estas declaraciones, reduciendo de hecho la hoja de estilo a:

```
h1 { color: red; }
p { color: blue; font-variant: small-caps }
em em { font-style: normal }
```

4.1.9 Comentarios

Los comentarios comienzan con los caracteres "/*" y terminan con los caracteres "*/". Pueden aparecer en cualquier lugar entre los comandos y su contenido no tiene ninguna influencia en el procesamiento. Los comentarios no pueden anidarse.

CSS también permite los delimitadores de comentarios de SGML ("<!--" y "-->") en determinados lugares, pero estos no delimitan comentarios CSS. Están permitidos para que las reglas de estilo que aparecen en un documento fuente HTML (en el elemento STYLE) puedan ocultarse en las aplicaciones del usuario anteriores a HTML 3.2. Ver la especificación HTML 4.0 ([\[HTML40\]](#)) para más información.

4.2 Reglas para el manejo de errores de análisis

En algunos casos, las aplicaciones del usuario deben ignorar parte de una hoja de estilo ilícita. Esta especificación utiliza el término ignorar para indicar que la aplicación del usuario analice la parte ilícita (con el fin de localizar su comienzo y su final) pero que por otro lado actúe como si allí no hubiera nada.

Para asegurarse que puedan agregarse en el futuro nuevas propiedades y nuevos valores a las propiedades existentes, se exige a las aplicaciones del usuario cumplir con las siguientes reglas cuando se encuentren con estos casos:

- **Propiedades desconocidas.** Las aplicaciones del usuario deben [ignorar](#) una [declaración](#) con una propiedad desconocida. Por ejemplo, si la hoja de estilo es:

```
h1 { color: red; rotation: 70minutes }
```

la aplicación del usuario tratará esto como si la hoja de estilo hubiera sido

```
h1 { color: red }
```

- **Valores ilícitos.** Las aplicaciones del usuario deben ignorar una declaración con un valor ilícito. Por ejemplo:

```
img { float: left } /* correcto en CSS 2.1 */
img { float: left here } /* "here" no es un valor de 'float' */
img { background: "red" } /* las palabras claves no puede ir entre
comillas */
img { border-width: 3 } /* debe especificarse el valor en las
unidades de medida */
```

Un analizador de CSS2.1 admitirá la primera regla e [ignorar](#) el resto, como si la hoja hubiera sido:

```
img { float: left }
img { }
img { }
img { }
```

Una aplicación del usuario que confirma una especificación de CSS futura puede aceptar también una o más de las otras reglas.

- **Declaraciones incorrectas.** Las aplicaciones del usuario deben manejar comandos inesperados encontrados mientras se lee hasta el final de la declaración, observando las reglas para mantener la concordancia de los pares (), [], {}, "", y ", y los escapes. Por ejemplo una declaración mal formulada puede omitir en la propiedad los dos puntos (:) o el valor.

Lo siguiente es todo equivalente:

```
p { color:green }
p { color:green; color } /* declaración incorrecta no tiene ':', ni valor */
p { color:red; color; color:green } /* igual que la anterior */
p { color:green; color: } /* declaración incorrecta no tiene valor */
p { color:red; color;; color:green } /* igual que la anterior */
p { color:green; color{;color:maroon} } /* comando incorrecto { } */
p { color:red; color{;color:maroon}; color:green } /* igual que la anterior */
*/
```

- **Palabras clave-arroba no válidas (ilícitas).** Las aplicaciones del usuario deben [ignorar](#) una palabra clave-arroba no válida junto con todo lo que le sigue hasta el punto y coma (;) o bloque ({...}) siguiente, incluyéndolo, cualquiera que sea el que aparezca primero. Por ejemplo, considere lo siguiente:

```
@three-dee {
  @background-lighting {
    azimuth: 30deg;
    elevation: 190deg;
  }
  h1 { color: red }
}
h1 { color: blue }
```

La regla-arroba '@three-dee' no forma parte de CSS2.1. Por consiguiente, la regla-arroba entera (hasta la tercer llave derecha, incluida esta) es [ignorada](#). Una aplicación del usuario CSS2.1 la [ignora](#), reduciendo de hecho la hoja de estilo a:

```
h1 { color: blue }
```

4.3 Valores

4.3.1 Enteros y números reales

Algunos tipos de valores pueden tener valores enteros (denotados por <entero>) o por valores con números reales (denotados por <número>). Los números reales y los enteros solo se especifican en notación decimal. Un <entero> consiste en uno o más dígitos de "0" a "9". Un <número> puede ser un <entero> o puede ser cero o más dígitos seguidos por un punto (.) seguido por uno o más dígitos. Ambos, enteros y números reales, pueden estar precedidos por un "-" o "+" para indicar su signo.

Advierta que muchas propiedades que permiten un entero o un número real como valor, en realidad restringen el valor a cierto rango, a menudo a un valor no negativo.

4.3.2 Medidas

Las medidas se refieren a las dimensiones horizontales y verticales.

El formato de un valor de medida (denotado por <medida> en esta especificación) es un signo opcional ('+' o '-', siendo '+' el predeterminado) seguido inmediatamente por un <número> (con o sin punto decimal) seguido inmediatamente por un identificador de la unidad (ej., px, deg, etc.). Después de la medida '0', el identificador de la unidad es opcional.

Algunas propiedades permiten valores de medida negativos, pero esto puede complicar el modelo del formato y pueden existir limitaciones propias de la implementación. Si un valor de medida negativo no es soportado, debe ser convertido al valor más cercano que pueda ser soportado.

Si se pone un valor negativo de medida en una propiedad que no permite valores de medida negativos, la declaración será ignorada.

Hay dos tipos de unidades de medida: relativas y absolutas. Las unidades de medidas relativas especifican una medida en relación a otra propiedad de medida. Las hojas de estilo que utilizan unidades relativas modificarán su escala de un medio a otro más fácilmente (ej., de un monitor de ordenador a una impresora láser).

Las unidades relativas son

- **em**: el tamaño ('font-size') de la fuente relevante
- **ex**: la 'altura de la x' de la fuente relevante
- **px**: píxeles, relacionado con los dispositivos visuales

```
h1 { margin: 0.5em }      /* em */
h1 { margin: 1ex }       /* ex */
p  { font-size: 12px }   /* px */
```

La unidad '**em**' es igual al valor computado de la propiedad 'font-size' del elemento en el que se usa. La excepción es cuando 'em' aparece en el valor de la propiedad 'font-size' en sí misma, en cuyo caso se refiere al tamaño de la fuente del elemento padre. Puede usarse para longitudes verticales u horizontales. (Esta unidad también es denominada quad-width en los textos sobre tipografía.)

La unidad '**ex**' es definida por la propiedad 'x-height' de la fuente. La altura de la x es denominada así debido a que es igual a la altura de la letra "x" minúscula. Sin embargo, 'ex' es definida aún para aquellas fuentes que no contienen la letra "x".

La regla

```
h1 { line-height: 1.2em }
```

significa que la altura de línea de los elementos h1 será un 20% mayor que el tamaño de la fuente de los elementos h1. Por otro lado:

```
h1 { font-size: 1.2em }
```

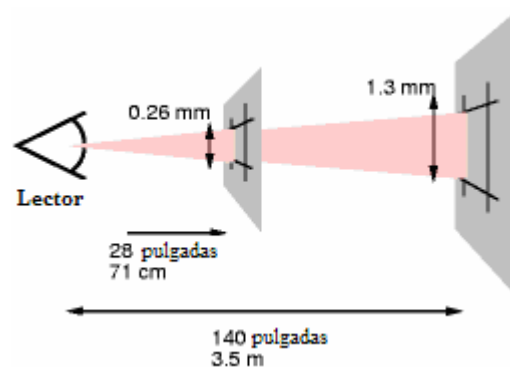
significa que la propiedad font-size de los elementos h1 será un 20% mayor que el tamaño de fuente heredado por los elementos h1.

Cuando se especifica en la raíz de la [estructura del documento](#) (Ej., "HTML" en HTML), 'em' y 'ex' se refieren al [valor inicial](#) de la propiedad.

Las unidades '**píxeles**' están relacionadas con la resolución de los dispositivos visuales, esto es, la mayoría de las veces un monitor de ordenador. Si la densidad de píxeles del dispositivo de salida es muy diferente de un típico monitor de ordenador la aplicación del usuario debe re-escalar los valores de los píxeles. Se recomienda que el píxel de referencia sea el ángulo visual de un píxel en un dispositivo con una densidad de píxeles de 90dpi y una distancia al lector del largo de su brazo. Para una longitud nominal del brazo de 28 pulgadas, el ángulo visual es de aproximadamente 0.0213 grados.

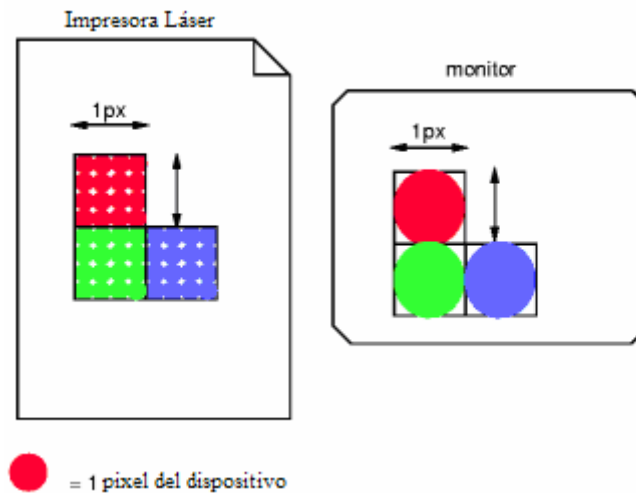
Para una distancia de lectura de la longitud de un brazo, 1 px corresponde aproximadamente a 0.26 mm (1/90 pulgadas). Cuando se imprime en una impresora láser, con una distancia de lectura un poco menor a la longitud del brazo (55 cm, 21 pulgadas), 1 px es aproximado a 0.20 mm. En una impresora de 300 puntos por pulgada (dpi), puede redondearse en 3 puntos (0.25 mm); en una impresora de 600 dpi, puede redondearse en 5 puntos.

Las dos imágenes de abajo ilustran el efecto de la distancia de lectura en el tamaño del píxel y el efecto de la resolución de un dispositivo. En la primer imagen, una distancia de lectura de 71 cm (28 pulgadas) resulta en un px de 0.26 mm, mientras que una distancia de lectura de 3.5 m (12 pies) requiere un px de 1.3 mm.



[D]

En la segunda imagen, un área de 1px por 1px es cubierta por un solo punto en un dispositivo de baja resolución (una pantalla de computadora), mientras que la misma área es cubierta por 16 puntos en un dispositivo de alta resolución (tal como una impresora láser de 400 dpi).



[D]

Los elementos hijos no heredan los valores relativos especificados para sus padres; ellos (generalmente) heredan los [valores computados](#).

```
body {
  font-size: 12px;
  text-indent: 3em; /* i.e., 36px */
}
h1 { font-size: 15px }
```

Las *unidades de medida absoluta* son útiles solamente cuando las propiedades físicas del medio de salida son conocidas. Las unidades absolutas son

- **in**: inches — 1 pulgada (inch) es igual a 2.54 centímetros.
- **cm**: centímetros
- **mm**: milímetros
- **pt**: puntos — los puntos usados por CSS2.1 equivalen a 1/72 pulgadas.
- **pc**: picas — 1 pica es igual a 12 puntos.

```
h1 { margin: 0.5in } /* inches (pulgadas)*/
h2 { line-height: 3cm } /* centímetros */
h3 { word-spacing: 4mm } /* milímetros */
h4 { font-size: 12pt } /* puntos */
h4 { font-size: 1pc } /* picas */
```

En caso de que la medida especificada no pueda ser [soportada](#), las aplicaciones del usuario deben aproximarla al [valor real](#).

4.3.3 Porcentajes

El formato de un valor en porcentaje (denotado por <porcentaje> en esta especificación) es un [número](#) seguido inmediatamente por '%'.</p>
</div>

Los valores expresados en porcentajes son siempre relativos a otro valor, por ejemplo una medida. Cada propiedad que admite porcentajes también define el valor al cual se refiere el porcentaje. El valor puede ser la de otra propiedad para el mismo elemento, una propiedad para un elemento antepasado o un valor en el contexto del formato (por ejemplo, el ancho de un [bloque de contenido](#)). Cuando un porcentaje es atribuido a una propiedad del elemento [raíz](#) y el porcentaje es definido en referencia al valor heredado de alguna propiedad, el valor resultante es la cantidad del porcentaje del [valor inicial](#) de esta propiedad.

```
p { font-size: 10px }
p { line-height: 120% } /* 120% de 'font-size' */
```

4.3.4 URL + URN = URI

Los URL (“Uniform Resource Locators”, Localizadores Uniformes de Recursos, ver [RFC1738](#) y [RFC1808](#)) proporcionan la dirección de un recurso en la Web. Una nueva forma en perspectiva para identificar recursos es llamada URN (“Uniform Resource Name”, Nombre Uniforme de Recurso). Juntos son llamados URI (“Uniform Resource Identifiers”, Identificadores Uniformes de Recursos, ver [URI](#)). Esta especificación usa el término URI.

Los valores de los URI en esta especificación son denotados por <uri>. La expresión funcional usada para designar los URI en los valores de la propiedad es “url()”, como en:

```
body { background: url("http://www.example.com/pinkish.png") }
```

El formato del valor de un URI es 'url(' seguido de un [espacio en blanco](#) opcional, seguido de una comilla simple (') o comilla doble (") opcional, seguida por el URI propiamente dicho, seguido de una comilla simple (') o comilla doble (") opcional, seguida por un espacio en blanco opcional, seguido de ')'. Las dos comillas tienen que ser las mismas.

Un ejemplo sin comillas:

```
li { list-style: url(http://www.example.com/redball.png) disc }
```

Los paréntesis, las comas, los espacios en blanco, las comillas simples (') y las comillas dobles (") que aparecen en un URI deben ir escapadas con una barra invertida: '\(', '\)', '\,', '\,'.

Dependiendo del tipo de URI, se podría escribir los caracteres de arriba como escapes en el URI (donde "(" = %28, ")" = %29, etc.) como se describe en [URI](#).

Con el fin de crear hojas de estilo modulares que no sean dependientes de la posición absoluta de un recurso, los autores pueden usar URI relativos. Los URI relativos (como se define en [RFC1808](#)) son convertidos a URI completos usando un URI de base. La RFC 1808, sección 3, define el algoritmo normativo para este proceso. Para las hojas de estilo CSS, el URI base es aquel de la hoja de estilo, no el del documento fuente.

Por ejemplo, suponga que la siguiente regla:

```
body { background: url("yellow") }
```

está localizada en una hoja de estilo designada por el URI

```
http://www.example.org/style/basic.css
```

El fondo de BODY en el documento fuente será un mosaico de cualquier imagen que sea descrita por el recurso designado por el URI

```
http://www.example.org/style/yellow
```

Las aplicaciones del usuario pueden variar el modo de manejar los URI que designan recursos no disponibles o inaplicables.

4.3.5 Contadores

Los contadores son denotados por identificadores (ver las propiedades 'counter-increment' y 'counter-reset'). Para referirse al valor de un contador, se usa la notación 'counter(<identificador>)' o 'counter(<identificador>, <estilo-de-lista>)'. El estilo predeterminado es 'decimal'.

Para referirse a una secuencia de contadores anidados del mismo nombre, la notación es 'counters(<identificador>, <cadena>)' o 'counters(<identificador>, <cadena>, <estilo-de-lista>)'. Ver "[Contadores anidados y área de alcance](#)" en el capítulo sobre [contenido generado](#).

En CSS2.1, el valor de los contadores sólo puede ser referenciado desde la propiedad 'content'. Note que 'none' es un posible <estilo-de-lista>: 'counter(x, none)' genera una cadena vacía.

Aquí hay una hoja de estilo que numera los párrafos (P) para cada capítulo (H1). Los párrafos son numerados con números romanos, seguidos por un punto y un espacio:

```
P {counter-increment: par-num}
H1 {counter-reset: par-num}
P:before {content: counter(par-num, upper-roman) ". "}
```

Los contadores que no están en el [área de alcance](#) de ningún 'counter-reset', se asume que han sido reiniciados a 0 por un 'counter-reset'.

4.3.6 Colores

Un <color> es o una palabra clave o una especificación numérica RGB.

La lista de los nombres de las palabras claves son: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, y yellow. Estos 17 colores tienen los siguientes valores:

maroon	#800000	red	#ff0000	orange	#ffa500	yellow	#ffff00	olive	#808000
purple	#800080	fuchsia	#ff00ff	white	#ffffff	lime	#00ff00	green	#008000
navy	#000080	blue	#0000ff	aqua	#00ffff	teal	#008080		
black	#000000	silver	#c0c0c0	gray	#808080				

Además de estas palabras clave de colores, los usuarios pueden especificar palabras clave que corresponden a los colores usados por ciertos objetos en el entorno del usuario. Por favor, consulte la sección sobre los [sistema de colores](#) para más información.

```
body {color: black; background: white }
h1 { color: maroon }
h2 { color: olive }
```

El modelo de color RGB es usado en la especificación numérica del color. Todos estos ejemplos especifican el mismo color:

```
em { color: #f00 }           /*#rgb*/
em { color: #ff0000 }       /*#rrggbb*/
em { color: rgb(255,0,0) }   /*rango de enteros 0 -255*/
em { color: rgb(100%, 0%, 0%) } /*rango flotante 0.0% - 100.0%*/
```

El formato de un valor RGB en notación hexadecimal es un '#' seguido inmediatamente por tres o seis caracteres hexadecimales. La notación RGB de tres dígitos (`#rgb`) es convertida en la forma de seis dígitos (`#rrggbb`) por repetición de dígitos, no por la adición de ceros. Por ejemplo, `#fb0` se expande a `#ffbb00`. Esto garantiza que el blanco (`#ffffff`) pueda ser especificado con la forma reducida (`#fff`) y quitar cualquier dependencia con la profundidad de color del monitor.

El formato de un valor RGB en la notación funcional es 'rgb(' seguido por una lista separada por comas de tres valores numéricos (ya sean tres valores enteros o tres valores de porcentajes) seguidos de ')'. El valor entero 255 corresponde a 100%, y a F o FF en la notación hexadecimal: `rgb(255,255,255) = rgb(100%,100%,100%) = #FFF`. [espacios en blanco](#) son permitidos alrededor de los valores numéricos.

Todos los colores RGB son especificados en el espacio de color sRGB (ver [\[SRGB\]](#)). Las aplicaciones del usuario pueden variar en la fidelidad con la que representan estos colores, pero el uso de sRGB proporciona una definición mensurable, objetiva y sin ambigüedades, de lo que debe ser el color, el cual puede ser relacionado con estándares internacionales (ver [\[COLORIMETRY\]](#)).

Las aplicaciones del usuario con conformidad pueden limitar su esfuerzo en el despliegue del color para realizar una corrección de gama en los mismos. sRGB especifica una gama de monitor de 2.2 bajo ciertas condiciones de percepción. Las aplicaciones del usuario deben ajustar los colores dados en CSS de manera tal que, en combinación con una gama de monitor "natural" en el dispositivo de salida, se produzca una gama de monitor efectiva de 2.2. Ver la sección sobre [corrección de gamma](#) para más detalles. Advierta que solo los colores especificados en CSS son afectados; ej. Se espera que las imágenes contengan su propia información de colores.

Los valores fuera de la gama del dispositivo deben ser recortados: los valores de rojo, verde y azul tienen que ser cambiados para adecuarlos al rango soportado por el dispositivo. Para un monitor CRT típico, cuya gama es la misma que sRGB, las tres reglas de abajo son equivalentes:

```
em { color: rgb(255,0,0) }           /* rango de enteros 0 - 255 */
em { color: rgb(300,0,0) }           /* recortado a rgb(255,0,0) */
em { color: rgb(255,-10,0) }         /* recortado a rgb(255,0,0) */
```

```
em { color: rgb(110%, 0%, 0%) } /* recortado a rgb(100%,0%,0%) */
```

Otros dispositivos, tales como impresoras, tiene gamas distintas a sRGB; algunos colores fuera del rango 0...255 de sRGB serán representables (dentro de la gama del dispositivo), mientras que otros colores dentro del rango 0...255 de sRGB estarán fuera de la gama del dispositivo y de ese modo serán recortados.

4.3.7 Cadenas

Las cadenas pueden escribirse con comillas dobles o con comillas simples. Las comillas dobles no pueden aparecer dentro de comillas dobles, a menos que sean escapadas (como "\" o como "\22"). Igualmente para las comillas simples ("\' o '\27')..

```
"ésta es una 'cadena'"
"ésta es una \"cadena\" "
'ésta es una "cadena"'
'ésta es una \'cadena\''
```

Una cadena no puede contener una nueva línea directamente. Para incluir una nueva línea en una cadena, use en escape representado por el carácter Unicode (U+000A); así como "\A" o "\0000a". Este carácter representa la noción genérica de "nueva línea" en CSS. Ver la propiedad ['content'](#) para un ejemplo.

Es posible separar las cadenas en varias líneas, por razones estéticas u otras razones, pero en tales casos la nueva línea tiene que ser escapada con la barra invertida (\). Por ejemplo, los siguientes dos selectores son exactamente lo mismo:

```
a[title="un título s\
o demasiado largo"] { /*...*/}
a[title="un título no demasiado largo "] { /*...*/}
```

4.3.8 Valores no soportados

Si una AU (Aplicación del Usuario) no soporta un valor en particular, esta debe ignorar ese valor cuando es pasado a la hoja de estilo, como si ese valor fuera un [valor ilícito](#). Por ejemplo:

```
h3 {
  display: inline;
  display: run-in;
}
```

Una AU que soporta el valor 'run-in' para la propiedad 'display', esta aceptará primero el valor del primer display, el cual será sobre escrito por el valor de la segunda declaración de display. Una AU (Aplicación del Usuario) que no soporte el valor 'run-in' procesará el primer valor de la declaración del display e ignorará el valor de la segunda declaración del display.

4.4 Representación de los documentos CSS

Una hoja de estilo CSS es una secuencia de caracteres del Conjunto Universal de Caracteres (Universal Character Set) (ver [\[ISO10646\]](#)). Para su transmisión y almacenamiento, estos caracteres deben ser codificados por una codificación de caracteres que soporte el conjunto de caracteres disponibles en US-ASCII (ej., ISO 8859-x, SHIFT JIS, etc.). Para una buena introducción a los juegos de caracteres y las codificaciones de caracteres, por favor consulte la especificación HTML 4.0 ([\[HTML40\]](#), capítulo 5). Ver también la especificación XML 1.0 ([\[XML10\]](#), secciones 2.2 y 4.3.3, y Apéndice F).

Cuando una hoja de estilo está incrustada en otro documento, tal como en el elemento STYLE o el atributo "style" de HTML, la hoja de estilo comparte la misma codificación de caracteres que todo el documento.

Cuando una hoja de estilo reside en un archivo separado, las aplicaciones del usuario deben observar las siguientes prioridades cuando determinan la codificación de caracteres de un documento (de la prioridad más alta a la más baja):

1. Un parámetro HTTP "charset" en un campo "Content-Type" (o parámetro similar en otro protocolo)
2. BOM y/o @charset (ver mas adelante)
3. <link charset=""> u otro metadata del mecanismo de linkeo
4. Mecanismos de referencia de las hojas de estilos o del documento
5. Asumir UTF-8

A lo sumo una regla @charset puede aparecer en una hoja de estilo externa y debe aparecer al comienzo del documento de la hoja de estilo, sin ser precedida por ningún carácter, a excepción de "BOM" (U+FEFF). Cualquier otra regla @charset debe ser ignorada por la AU (Aplicación del Usuario).

Después de "@charset", los autores especifican el nombre de una codificación de caracteres. Por ejemplo:

```
@charset "ISO-8859-1";
```

El nombre debe ser un nombre del conjunto descrito en el registro IANA (Ver [\[CHARSETS\]](#) para una lista del conjunto de caracteres.

Esta especificación no establece que codificaciones de caracteres deben soportar las aplicaciones del usuario.

Esta especificación no especifica que algoritmos debe aplicar una AU (Aplicación del Usuario) derivada de la codificación para el BOM y el @charset. En particular, no especifica que codificación utilizar si BOM y @charset tienen conflictos. Esto se tiene previsto definen en CSS3.

Note que la confianza en la estructura de @charset teóricamente plantea un problema debido a que no hay información a priori sobre cómo está codificada. En la práctica, sin embargo, las codificaciones de uso generalizado en Internet están basadas en ASCII, UTF-16, UCS-4, o (raramente) en EBCDIC. Esto significa que, en general, los valores del byte inicial de un documento habilitan a la aplicación del usuario para detectar la familia de codificación fidedignamente, lo cual provee suficiente información para decodificar la regla @charset, la que a su vez determina la codificación de caracteres exacta. 4.4.1 Referencias a caracteres no

4.4.1 Referencia a caracteres no representados en una codificación de caracteres

Una hoja de estilo puede tener que referirse a caracteres que no pueden ser representados en la actual codificación de caracteres. Estos caracteres deben ser escritos como referencias [escapadas](#) a los caracteres ISO 10646. Estos escapes sirven al mismo propósito que los caracteres numéricos referenciados en los documentos HTML o XML (ver [HTML40](#), capítulo 5 y 25).

El mecanismo de escape de caracteres debería ser utilizado cuando solamente unos cuantos caracteres deben ser representados de esta forma. Si la mayor parte del documento requiere de escapes los autores deben codificarlo con la codificación mas apropiada (Ej., Si el documento contiene una gran cantidad de caracteres en griego los autores podrían usar "ISO-8859-7" o "UTF-8").

Los procesadores intermedios que utilizan una codificación de caracteres diferente pueden traducir estas secuencias con escape a secuencias de bytes de esa codificación. Los procesadores intermedios no deben, por otro lado, alterar las secuencias de escape que cancelen el significado especial de un carácter en ASCII.

Las [aplicaciones del usuario con conformidad](#) deben asignar en Unicode a todos los caracteres en cualquier codificación de caracteres que reconozca (o debe comportarse como si lo hubiera hecho).

Por ejemplo, un documento transmitido como ISO-8859-1 (Latin-1) no puede contener directamente las letras griegas: "" (griego: "kouros") tiene que escribirse como "\3BA\3BF\3C5\3C1\3BF\3C2".

Nota. En HTML 4.0, las referencias con caracteres numéricos son interpretadas en los valores del atributo "style" pero no en el contenido del elemento STYLE. Debido a esta asimetría, recomendamos que los autores utilicen el mecanismo de escape de caracteres de CSS en lugar de referencias con caracteres numéricos para ambos, el atributo "style" y el elemento STYLE. Por ejemplo, recomendamos:

```
<SPAN style="font-family: L\FC beck">...</SPAN>
```

En lugar de:

```
<SPAN style="font-family: L&#252;beck">...</SPAN>
```

5 Selectores

Contenido

- [5.1 Equivalencia de patrones](#)
- [5.2 Sintaxis del selector](#)
 - [5.2.1 Agrupamiento](#)
- [5.3 Selector universal](#)
- [5.4 Selectores de tipos](#)
- [5.5 Selectores de descendiente](#)
- [5.6 Selectores de hijos](#)
- [5.7 Selectores de hermanos adyacentes](#)
- [5.8 Selectores de atributos](#)
 - [5.8.1 Atributos equivalentes y valores de atributos](#)
 - [5.8.2 Valores predeterminados de los atributos en la DTDs](#)
 - [5.8.3 Selectores de clase](#)
- [5.9 Selectores ID](#)
- [5.10 Pseudo-elementos y pseudo-clases](#)
- [5.11 Pseudo-clases](#)
 - [5.11.1 La pseudo-clase :first-child](#)
 - [5.11.2 Las pseudo-clases vínculo :link y :visited](#)
 - [5.11.3 Las pseudo-clases dinámicas: :hover :active y :focus](#)
 - [5.11.4 La pseudo-clase del lenguaje: :lang](#)
- [5.12 Pseudo-elementos](#)
 - [5.12.1 El pseudo-elemento :first-line](#)
 - [5.12.2 El pseudo-elemento :first-letter](#)
 - [5.12.3 Los pseudo-elementos :before y :after](#)

5.1 Equivalencia de patrones

En CSS, las reglas de equivalencias de patrones determinan qué reglas del estilo se aplican a los elementos en la [estructura del documento](#). Estos patrones, llamados selectores, pueden ir desde el simple nombre de un elemento hasta complejos patrones contextuales. Si todas las condiciones en el patrón son verdaderas para cierto elemento, el selector *equivale* al elemento.

La distinción entre mayúsculas/minúsculas (case-sensitivity) en los nombres de los elementos de los selectores, dentro del lenguaje del documento, depende del lenguaje del documento. Por ejemplo, en HTML, los nombres de los elementos son insensibles a la diferencia entre mayúsculas/minúsculas (case-insensitive), pero en XML sí hacen distinción entre mayúsculas y minúsculas (case-sensitive).

La tabla siguiente resume la sintaxis de los selectores de CSS 2.1:

Patrón	Significado	Descrito en la sección
*	Equivale a cualquier elemento.	Selector universal
E	Equivale a cualquier elemento de E (ej., un elemento del tipo E).	Selectores de tipos
E F	Equivale a cualquier elemento de F que sea un descendiente de un elemento de E.	Selectores de descendientes
E > F	Equivale a cualquier elemento de F que sea hijo de un elemento E.	Selectores de hijos
E:first-child	Equivale al elemento E cuando E es el primer hijo de su padre.	La pseudo-clase de:first-child
E:link E:visited	Equivale al elemento E si E es el ancla de origen de un hipervínculo cuyo destino aún no se ha visitado (:link) o que ya se visitó (:visited).	Las pseudo-clases :link
E:active E:hover E:focus	Equivale a E durante ciertas acciones del usuario.	Las pseudo-clases dinámicas
E:lang(c)	Equivale a un elemento de tipo E si está en el lenguaje (humano) c (el lenguaje del documento especifica cómo se determina el idioma).	la pseudo-clase :lang()
E + F	Equivale a cualquier elemento de F precedido inmediatamente por un elemento E hermano.	Selectores adyacentes
E[foo]	Equivale a cualquier elemento E con el atributo "foo" (cualquier valor).	Selectores de atributo
E[foo="warning "]	Equivale a cualquier elemento de E cuyo valor del atributo "foo" sea exactamente igual a "warning".	Selectores de atributos
E [foo~="warning "]	Equivale a cualquier elemento E cuyo atributo "foo" tiene un valor consistente en una lista de valores separados por espacios, uno de los cuales es exactamente igual a "warning".	Selectores de atributos
E[lang = "en"]	Equivale a cualquier elemento E cuyo atributo "lang" tiene una lista de valores separados por guiones que comienzan (desde la izquierda) con "en".	Selectores de atributos
DIV.warning	<i>Específico del lenguaje.</i> (en HTML, igual que DIV[class~="warning "].)	Selectores de clase
E#myid	Equivale a cualquier elemento de E cuyo ID es igual a "myid".	Selectores de ID

5.2 Sintaxis de los selectores

Un *selector simple* es tanto un [selector de tipos](#) como un [selector universal](#) seguido inmediatamente por cero o más [selectores de atributos](#), [selectores de ID](#), o [pseudo-clases](#), en cualquier orden. El selector simple es equivalente si todos sus componentes son equivalentes.

Nota: la terminología usada aquí en CSS 2.1 es diferente de la que se utiliza en CSS3. Por ejemplo, un "selector simple" en CSS3 se refiere a una parte pequeña de un selector de CSS 2.1. Vea el módulo de los selectores CSS3 [[CSS3SEL](#)].

Un *selector* es una cadena de uno o más selectores simples separados por combinadores. Los combinadores son: espacio en blanco, ">", y "+". El espacio en blanco puede aparecer entre un combinador y los selectores simples que están próximos a él.

Los elementos de la estructura del documento que equivalen a un selector son llamados **sujetos del selector**. Un selector que consiste en un solo selector simple se corresponde con cualquier elemento que satisface sus requisitos. Anteponiendo un selector simple y un combinador a una cadena se imponen restricciones adicionales para la equivalencia, de este modo los sujetos de un selector son siempre un sub-conjunto de los elementos que equivalen al último selector simple.

Un [pseudo-elemento](#) se puede añadir al selector simple pasado en una cadena, en cuyo caso la información del estilo se aplica a una sub-parte de cada sujeto

5.2.1 Agrupamiento

Cuando varios selectores comparten las mismas declaraciones, pueden ser agrupados en una lista separada por comas.

En este ejemplo, condensamos tres reglas con declaraciones idénticas en uno. Así,

```
H1 { font-family: sans-serif }
H2 { font-family: sans-serif }
H3 { font-family: sans-serif }
```

es equivalente a:

```
H1, H2, H3 { font-family: sans-serif }
```

CSS ofrece también otros mecanismos "resumidos", incluyendo [declaraciones múltiples](#) y [propiedades resumidas](#).

5.3 Selector universal

El selector universal, representado por "*", se corresponde con el nombre de cualquier tipo de elemento. Es equivalente con cualquier elemento en la [estructura del documento](#)

Si el selector universal no es el único componente de un [selector simple](#), el "*" puede ser omitido. Por ejemplo:

- `*[lang=fr]` y `[lang=fr]` son equivalentes.
- `*warning` y `warning` son equivalentes.
- `*#myid` y `#myid` son equivalentes.

5.4 Selectores de tipos

Un selector de tipo se corresponde con el nombre de un tipo de elemento en el lenguaje del documento. Un selector de tipo equivale con cada instancia del tipo de elemento en la estructura del documento.

La regla siguiente engloba a todos los elementos H1 de la estructura del documento:

```
H1 { font-family: sans-serif }
```

5.5 Selectores de descendiente

A veces, los autores pueden querer selectores que se correspondan con un elemento que es el descendiente de otro elemento en la estructura del documento (ej., "Que equivalga a aquellos elementos EM que están contenidos en un elemento H1"). Los selectores de descendientes expresan tal relación dentro de un patrón. Un selector de descendiente se compone de dos o más selectores separados por un [espacio en blanco](#). Un selector de descendiente de la forma "A B" engloba a B cuando este es descendiente de algún elemento [antepasado](#) de A.

Por ejemplo, considere las reglas siguientes:

```
H1 { color: red }
EM { color: red }
```

Aunque la intención de estas reglas es agregar énfasis al texto cambiando su color, el efecto será perdido en un caso como:

```
<H1>El título es <EM>muy</EM> importante</H1>
```

Tratemos este caso supliendo las reglas anteriores con una regla que fije el color del texto al azul siempre que un EM ocurra en cualquier lugar dentro de un H1:

```
H1 { color: red }
EM { color: red }
H1 EM { color: blue }
```

La tercera regla emparejará el EM en el fragmento siguiente:

```
<H1>Este <SPAN class="myclass">título es <EM>muy</EM>
importante</SPAN></H1>
```

El selector siguiente:

```
div * p
```

empareja un elemento de P que sea un nieto o un descendiente más lejano de un elemento DIV. Observe el uso opcional del espacio en blanco a ambos lados del "*" que no forma parte del selector universal; el espacio en blanco es un combinador que indica que el DIV debe ser el antepasado de un cierto elemento, y que ese elemento debe ser un antepasado de P.

El selector en la regla siguiente, que combina los selectores descendiente con los selectores de atributos, empareja cualquier elemento que (1) tiene asignado el atributo "href" y (2) está dentro de un P que a su vez está dentro de un DIV:

```
div p * [ href ]
```

5.6 Selectores hijos

Un selector hijo engloba a un elemento que es hijo de un cierto elemento. Un selector hijo se compone de dos o más selectores separados por ">".

La regla siguiente fija el estilo de todos los elementos de P que sean hijos de BODY:

```
BODY > P { line-height: 1.3 }
```

El ejemplo siguiente combina los selectores descendientes y los selectores hijo:

```
DIV OL>LI P
```

Equivale a un elemento P que sea un descendiente de un LI; el elemento LI debe ser el hijo de un elemento OL; el elemento OL debe ser un descendiente de un DIV. Observe que el espacio en blanco opcional alrededor del combinador ">" se ha descartado.

Para información acerca de la selección del primer hijo de un elemento, por favor ver la sección acerca de la pseudos-clase [:first-child](#) abajo.

5.7 Selectores de hermanos adyacentes

Los selectores de hermanos adyacentes tienen la sintaxis siguiente: E1 + E2, donde E2 es el sujeto del selector. El selector equivale si E1 y E2 comparten el mismo padre en la estructura del documento y E1 precede inmediatamente a E2, ignorando los nodos no-element (tales como nodos de texto y comentarios).

Así, la regla siguiente indica que cuando un elemento P sigue inmediatamente a un elemento MATH, no debe tener sangría:

```
MATH + P { text-indent: 0 }
```

El ejemplo siguiente reduce el espacio vertical que separa un H1 y un H2 que lo sigue inmediatamente:

```
H1 + H2 { margin-top: -5mm }
```

La regla siguiente es similar a la del ejemplo anterior, excepto que agrega un selector de clase. Así, el formato especial ocurre solamente cuando H1 tiene especificado `class="opener"`

```
H1.opener + H2 { margin-top: -5mm }
```

5.8 Selectores de atributos

CSS 2.1 permite que los autores especifiquen las reglas que sean equivalentes con los atributos de los elementos definidos en el documento fuente.

5.8.1 Atributos equivalentes y valores de atributos

Los selectores de atributos pueden equivaler de cuatro maneras:

[att]

Equivale con los elementos que tiene asignado el atributo "att", cualquiera que sea el valor del atributo.

[att=val]

Equivale cuando el valor del atributo "att" del elemento es exactamente "val".

[att~val]

Equivale cuando el valor del atributo "att" del elemento es una lista de "palabras" separadas por espacios, y una de las cuales es exactamente "val". Si se utiliza este selector, las palabras en el valor no deben contener espacios (puesto que son separadas por espacios).

[att|=val]

Equivale cuando el valor del atributo "att" del elemento es una lista de "palabras" separadas por guiones, comenzando con "val". La correspondencia siempre comienza al principio del valor del atributo. Esto está pensado principalmente para permitir equivalencias con el sub-código del lenguaje (Ej., el atributo "lang" en HTML) según lo descrito en RFC 3066 ([RFC3066](#)).

Los valores del atributo deben ser identificadores o cadenas. La distinción entre mayúsculas/minúsculas (case-sensitivity) en los nombres y valores de atributos de los selectores depende del lenguaje del documento.

Por ejemplo, el selector siguiente del atributo equivale a todos los elementos H1 que especifiquen el atributo "título", cualquiera que sea su valor:

```
H1[title] { color: blue; }
```

En el ejemplo siguiente, el selector equivale (engloba) a todos los elementos SPAN cuyo atributo "class" tenga exactamente el valor "ejemplo":

```
SPAN[class=ejemplo] { color: blue; }
```

Los selectores múltiples de atributos pueden ser utilizados para referirse a varios atributos de un elemento, o incluso varias veces al mismo atributo.

Aquí, el selector equivale a todos los elementos del SPAN cuyo atributo "hola" tiene exactamente el valor "Cleveland" y cuyo atributo "adiós" tiene exactamente el valor "Columbus":

```
SPAN[hola="Cleveland"][adios="Columbus"] { color: blue; }
```

Los selectores siguientes ilustran las diferencias entre "=" y "~=". El primer selector equivale, por ejemplo, al valor "copyright copyleft copyeditor" para el atributo "rel". El segundo selector solo será equivalente cuando el atributo "href" tiene el valor "http://www.w3.org/".

```
a[rel~="copyright "]
a[href="http://www.w3.org/" ]
```

La regla siguiente oculta todos los elementos para los cuales el valor de la cualidad "lang" sea "fr" (es decir, la lengua es francés).

```
*[LANG=fr] { display : none }
```

La siguiente regla será equivalente a valores del atributo "lang" que empiecen con "en", incluyendo "en", "en-US" y "en-cockney":

```
*[LANG|= "en"] { color : red }
```

De modo similar, la siguiente regla de una hoja de estilo auditiva permite leer un escrito en voz alta con voces diferentes para cada rol:

```
DIALOGUE[character=romeo]
  { voice-family: "Laurence Olivier", charles, male }

DIALOGUE[character=juliet]
  { voice-family: "Vivien Leigh", victoria, female }
```

5.8.2 Valores predeterminados de los atributos en la DTDs

Las correspondencias tienen lugar en los valores de los atributos en la estructura del documento. Para lenguajes de documento fuera de HTML, los valores predeterminados de los atributos pueden definirse en una DTD (Document Type Definition, Definición del tipo de documento) o en cualquier otra parte. Las hojas de estilo deben diseñarse de manera que trabajen aún cuando los valores predeterminados no estén incluidos en la estructura del documento.

Con mayor precisión, una AU (Aplicación del Usuario) no requiere leer un "subconjunto externo" de DTD pero si requiere buscar el valor predeterminado (por defecto) del atributo en el "subconjunto interno" del documento (Ver [XML10](#) para la definición de estos subconjuntos.)

Una AU (Aplicación del Usuario) que reconoce un namespace en XML [\[XMLNAMESPACES\]](#) no necesita utilizar este conocimiento del namespace para saber el valor predeterminado del atributo como si este estuviera en el documento

Nota Usualmente, en la práctica se elige ignorar los subconjuntos externos.

Por ejemplo, considere un elemento EJEMPLO con un atributo "notación" que tiene un valor predeterminado como "decimal". El fragmento DTD podría ser:

```
<!ATTLIST EJEMPLO notación (decimal, octal) "decimal">
```

Si la hoja de estilo contiene las reglas

```
EJEMPLO[notación=decimal] { /*... opciones de la propiedad predeterminada ...*/ }  
EJEMPLO[notación=octal] { /*... otras opciones ...*/ }
```

entonces para tomar los casos en los cuales este atributo es establecido por defecto y no explícitamente, podría agregarse la siguiente regla:

```
EJEMPLO { /*... opciones de la propiedad predeterminada ...*/ }
```

Debido a que este selector es menos específico que un selector de atributo, deberá ser usado para el caso predeterminado. Debe tomarse la precaución de que todos los otros valores del atributo que no tengan el mismo estilo que el predeterminado sean explícitamente cubiertos.

5.8.3 Selectores de clases

Trabajando con HTML, los autores pueden utilizar el punto (.) como una alternativa a “~=” cuando la correspondencia sea con el atributo "class". Así, en HTML, "DIV.valor" y "DIV[class~=valor]" tiene el mismo significado. El valor del atributo debe seguir inmediatamente al ".". Las Aus (Aplicaciones de Usuarios) pueden aplicar selectores utilizando la notación punto (.) en documentos XML si la AU (aplicación del usuario) tiene especificado namespace conocidos que determinen que atributos son las “clases” de los respectivos atributos namespace Un ejemplo de namespace específicos conocidos es el proceso en la especificación para un namespace particular (Ej.. SVG 1.0 [\[SVG10\]](#) describe el [atributo SVG "class"](#) y como una AU puede interpretarlo, igualmente MathML 2.0 [\[MATH20\]](#) describe el [atributo MathML "class"](#).)

Por ejemplo, podemos asignar información de estilo a todos los elementos con class~="pastoral" como sigue:

```
*.pastoral { color: green } /* todos los elementos con class~=pastoral */
```

o simplemente

```
.pastoral { color: green } /* todos los elementos con class~=pastora */
```

El siguiente sólo asigna estilo a los elementos H1 con class~="pastoral":

```
H1.pastoral { color: green } /* elementos H1 con class~=pastoral */
```

Dadas estas reglas, la primera instancia de H1 abajo no tendría el texto en verde, mientras que la segunda sí:

```
<H1>Verde, no</H1>  
<H1 class="pastoral">Muy verde</H1>
```

Para equivaler a un subconjunto de valores "class", cada valor debe ir precedido por un ".", en cualquier orden.

Por ejemplo, la siguiente regla equivale a cualquier elemento P a cuyo atributo "class" le ha sido asignado una lista de valores separados por espacios que incluyen "pastoral" y "marino":

```
P.pastoral.marino { color: green }
```

Esta regla equivale en el caso de class="pastoral rojo azul marino" pero no se corresponde en class="pastoral rojo".

Nota. CSS le otorga tanto poder al atributo "class", que los autores podrían conceptualmente diseñar su propio "lenguaje de documento" basados en elementos con casi ninguna presentación asociada (tales como DIV y SPAN en HTML) y asignando información acerca del estilo a través del atributo "class". Los autores deben evitar esta práctica ya que los elementos estructurales de un lenguaje de documento comúnmente tienen significados aceptados y reconocidos mientras que las clases definidas por el autor pueden no tenerlos.

5.9 Selectores de ID

Los lenguajes del documento pueden contener atributos que se declaran con el tipo ID. Lo que hace especial a un atributo del tipo ID es que dos de tales atributos no pueden tener el mismo valor; cualquiera que sea el lenguaje del documento, un atributo ID puede ser usado para identificar a su elemento de manera excluyente. En HTML todos los atributos ID son denominados "id"; las aplicaciones XML pueden nombrar de manera diferente a los atributos ID, pero se aplica la misma restricción.

El atributo ID de un lenguaje de documento le permite a los autores asignar un identificador a una instancia del elemento dentro de la estructura del documento. Los selectores ID de CSS equivalen a una instancia del elemento basado en su identificador. Un selector ID de CSS contiene un "#" seguido inmediatamente por el valor de ID.

Nota CSS no especifica como una AU (Aplicación de Usuario) conoce los atributos del ID de un elemento. La AU (Aplicación de Usuario) puede por ejemplo leer el DTD del documento, obtener la información del código fuerte o preguntar al usuario.

El siguiente selector de ID equivale al elemento H1 cuyo atributo ID tiene el valor "capitulo1":

```
h1#capitulo1 { text-align: center }
```

En el siguiente ejemplo, la regla de estilo equivale al elemento que tiene el valor de ID "z98y". La regla será, de este modo, equivalente al elemento P:

```

<HEAD>
  <TITLE>Equivalencia con P</TITLE>
  <STYLE type="text/css">
    *#z98y { letter-spacing: 0.3em }
  </STYLE>
</HEAD>
<BODY>
  <P id=z98y>texto espaciado</P>
</BODY>

```

En el ejemplo que sigue, sin embargo, la regla de estilo sólo equivale a un elemento H1 que tiene un valor ID de "z98y". La regla no será equivalente con el elemento P en este ejemplo:

```

<HEAD>
  <TITLE>Equivalencia con H1 solamente</TITLE>
  <STYLE type="text/css">
    H1#z98y { letter-spacing: 0.5em }
  </STYLE>
</HEAD>
<BODY>
  <P id=z98y>Texto espaciado</P>
</BODY>

```

Los selectores de ID tienen una precedencia mayor que los selectores de atributos. Por ejemplo, en HTML, el selector #p123 es más específico que [ID=p123] en términos de [cascada](#).

Nota. En XML 1.0 [XML10], la información acerca de qué atributos contiene el ID de un elemento está contenida en una DTD. Cuando analiza XML, las AU no siempre leen la DTD, y así no pueden saber qué es el ID de un elemento. Si el diseñador de una hoja de estilo conoce o sospecha que éste será el caso, deberá usar en cambio los selectores de atributos normales: [name=p371] en lugar de #p371. Sin embargo, el orden en la cascada de los selectores de atributos normales es diferente de los selectores de ID. Puede llegar a ser necesario agregar una prioridad "!important" a las declaraciones: [name=p371] {color: red !important}. Por supuesto, los elementos de los documentos XML 1.0 sin una DTD no tiene ID en absoluto.

5.10 Pseudo-elementos y pseudo-clases

En CSS2.1, el estilo se atribuye a un elemento basado en su posición dentro de [estructura del documento](#). Este sencillo modelo es suficiente para la mayoría de los casos, pero algunos temas comunes en la edición pueden no ser posibles por medio de la [estructura del documento](#). Por ejemplo en HTML 4.0 (Ver [HTML40]), ningún elemento se refiere a la primera línea de un párrafo y, por consiguiente, ningún selector CSS simple puede referirse a ella.

CSS introduce los conceptos de pseudo-elementos y pseudo-clases para permitir aplicar el formato basado en la información que está fuera de la estructura del documento.

- Los **pseudo-elementos** crean abstracciones acerca de la estructura del documento más allá de aquellas especificadas por el lenguaje del documento. Por ejemplo, los lenguajes de documento no ofrecen mecanismos para acceder a la primera letra o a la primera línea del contenido de un elemento. Los pseudo-elementos de CSS permiten a los diseñadores de hojas de estilo referirse a esta información inaccesible por otros medios.

Los pseudo-elementos también pueden proporcionar a los diseñadores de hojas de estilo un modo de asignar estilos a un elemento que no existe en el documento fuente (ej., los pseudo-elementos [:before](#) y [:after](#) permite acceder al contenido generado).

- Las **pseudo-clases** clasifican a los elementos basado en propiedades más allá de su nombre, atributos o contenidos; en principio, atributos que no pueden deducirse de la estructura del documento. Las pseudo-clases pueden ser dinámicas, en el sentido de que un elemento puede adquirir o perder una pseudo-clase a medida que el usuario interactúa con el documento. La excepción es [':first-child'](#), que puede deducirse de la estructura del documento, y [':lang\(\)'](#), que puede deducirse de la estructura del documento en algunos casos.

Ni los pseudo-elementos ni las pseudo-clases aparecen en el documento fuente o en la estructura del documento.

Las pseudo-clases son permitidas en cualquier lugar dentro de un selector mientras que los pseudo-elementos sólo pueden aparecer después del [sujeto](#) del selector.

Los nombres de los pseudo-elementos y las pseudo-clases no hacen distinción entre mayúsculas y minúsculas (case-insensitive).

Algunas pseudo-clases son mutuamente excluyentes, mientras otras pueden aplicarse simultáneamente al mismo elemento. En caso de reglas conflictivas, el [orden de cascada](#) normal determina el resultado.

5.11 Pseudo-clases

5.11.1 La pseudo-clase [:first-child](#)

La pseudo-clase [:first-child](#) equivale a un elemento que es el primer hijo de cualquier otro elemento.

En el ejemplo siguiente, el selector equivale a cualquier elemento P que sea el primer hijo de un elemento DIV. La regla elimina el sangrado del primer párrafo de un DIV:

```
div > p:first-child { text-indent: 0 }
```

Este selector sería equivalente a P dentro de DIV en el siguiente fragmento:

```
<P> El ultimo P antes de la nota
<DIV class="nota">
  <P> El primer P dentro de la nota
</DIV>
```

pero no se correspondería con el segundo P en el siguiente fragmento:

```
<P> El último P antes de la nota
<DIV class="nota">
  <H2>Note</H2>
  <P> El primer P dentro de la nota
</DIV>
```

La siguiente regla especifica 'bold' para el peso de la fuente de cualquier elemento EM que es algún descendiente de un elemento P que es el primer hijo:

```
p:first-child em { font-weight : bold }
```

Observe que las cajas [anónimas](#) no forman parte de la estructura del documento, no se cuentan en el cálculo del primer hijo.

Por ejemplo, el EM en:

```
<P>abc <EM>default</EM>
```

es el primer hijo de P.

Los siguientes dos selectores son equivalentes:

```
* > A:first-child /* A es el primer hijo de cualquier elemento */  
A:first-child /* Igual */
```

5.11.2 Las pseudo-clases: :link y :visited

Las aplicaciones del usuario normalmente muestran los vínculos no visitados de un modo diferenciado de aquellos previamente visitados. CSS proporciona las pseudo-clases ':link' y ':visited' para distinguirlos:

- La **pseudo-clase :link** se aplica a los vínculos que aún no han sido visitados.
- La **pseudo-clase :visited** se aplica una vez que el vínculo ha sido visitado por el usuario

Nota. Después de transcurrir cierto tiempo, las aplicaciones del usuario pueden elegir devolver un vínculo visitado a la condición de (no visitado) ':link'.

Los dos estados son mutuamente excluyentes.

El lenguaje del documento determina qué elementos forman parte de la matriz de origen de los hipervínculos. Por ejemplo, en HTML 4.0, las pseudo-clases link se aplica a los elementos A con el atributo "href". De este modo, las siguientes dos declaraciones CSS2.1 tienen un efecto similar:

```
a:link { color: red }  
:link { color: red }
```

Si el siguiente :

```
<A class="externo" href="http://out.side/">link externo </A>
```

Ha sido visitado, esta regla:

```
a.externo:visited { color: blue }
```

provocará que sea azul.

Nota. Es posible que los autores de hojas de estilo abusen de las pseudo-clases `:link` y `:visited` para determinar que sitios han sido visitados por el usuario sin su consentimiento.

Las AUs (Aplicaciones de Usuarios) pueden conocer todos los links no visitados o implementar otros medios para preservar la privacidad de los usuarios mientras diferencien los links visitados y no visitados . Ver [\[P3P\]](#) para mas información sobre privacidad.

5.11.3 Las pseudo-clases dinámicas: `:hover`, `:active`, y `:focus`

Las aplicaciones del usuario interactivas a veces cambian el procesamiento en respuesta a las acciones del usuario. CSS proporciona tres pseudo-clases para los casos comunes:

- La **pseudo-clase `:hover`** se aplica mientras el usuario señala un elemento (con algún dispositivo para apuntar), pero no lo activa. Por ejemplo, una aplicación del usuario visual puede aplicar esta pseudo-clase cuando el cursor (el puntero del ratón) se mueve por encima de una caja generada por el elemento. Las aplicaciones del usuario que no ofrecen soporte a los [medios interactivos](#) no tienen que apoyar esta pseudo-clase. Algunas aplicaciones del usuario con conformidad que soportan [medios interactivos](#) pueden no ser adecuadas para soportar esta pseudo-clase (ej., lápices ópticos).
- La **pseudo-clase `:active`** se aplica mientras un elemento está siendo activado por el usuario. Por ejemplo, el lapso durante el cual el usuario presiona el botón del ratón y lo suelta.
- La **pseudo-clase `:focus`** se aplica mientras un elemento tiene el foco (acepta eventos del teclado u otras formas de entrada de texto).

Estas pseudo-clases no son mutuamente excluyentes. Un elemento puede recibir varias de ellas al mismo tiempo.

CSS no define qué elementos pueden estar en los estados anteriores o cómo se ingresa y se sale de los estados. Mediante el uso de scripts se puede decidir si los elementos reaccionan a los eventos del usuarios o no, y los distintos dispositivos y aplicaciones del usuario pueden tener diferentes maneras de apuntar o activar los elementos.

CSS2.1 no define si el elemento padre está `:active` o `:hover` el hijo también tiene ese estado.

Las aplicaciones del usuario no son obligadas a redibujar el documento en uso a causa de las transiciones de las pseudo-clases. Por ejemplo, una hoja de estilo puede especificar que la propiedad 'font-size' de un vínculo `:active` debe ser mayor que la de un vínculo inactivo, pero como esto puede ocasionar que las letras cambien de posición cuando el lector selecciona el vínculo, una AU puede ignorar la regla de estilo correspondiente.

```
A:link      { color: red }      /* vínculos no visitados */
A:visited  { color: blue }   /* vínculos visitados */
A:hover    { color: yellow } /* el usuario señala el vínculo */
A:active   { color: lime }   /* vínculos activos */
```

Observe que A:hover debe ir después que las reglas A:link y A:visited, de otro modo las reglas de cascada ocultarán la propiedad 'color' de la regla A:hover. También, debido a que A:active está ubicada después de A:hover, el color activo (lime) será aplicado cuando el usuario active y señale el elemento A.

Un ejemplo de combinación de pseudo-clases dinámicas:

```
a:focus { background: yellow }
a:focus:hover { background: white }
```

El último selector equivale a los elementos A que se encuentran en la pseudo-clase :focus y en la pseudo-clase :hover.

Para información acerca de la representación del foco en los contornos, consulte por favor la sección sobre [foco dinámico en los contornos](#).

Nota. En CSS1, la pseudo-clase ':active' era mutuamente excluyente con ':link' y ':visited'. Este ya no es el caso. Un elemento puede ser ambos, ':visited' y ':active' (o ':link' y ':active') y la reglas normales de cascada determinan qué propiedades se aplican.

Nota. Observe así que en CSS1, la pseudo-class ':active' solo se aplicaba a links.

5.11.4 La pseudo-clase de lenguaje: :lang

Si el lenguaje del documento especifica cómo se determina el lenguaje humano de un elemento, es posible escribir selectores en CSS que coincidan con elementos basados en su lenguaje. Por ejemplo, en HTML [\[HTML40\]](#), el lenguaje se determina por una combinación del atributo "lang", el elemento META y probablemente por la información del protocolo (tal como en los encabezados HTTP). XML utiliza un atributo denominado xml:lang, y puede haber otros métodos específicos del lenguaje del documento para determinar el lenguaje.

La pseudo-clase ':lang(C)' equivale si el elemento está en lenguaje C. Si hay una equivalencia basada solamente en el identificador C que sea igual o no o una subcadena separada por un guión, del valor de lenguaje del elemento; de la misma manera que si fuera realizado por el operador '≡'. El identificador de C no tiene que ser un nombre de lenguaje válido.

excepción: C puede ser abierto, pero no está definido en CSS2.1 que es equivalente en ese caso (Esto será definido en CSS3)

Nota: Es recomendable, que los documentos y los protocolos indicados en el lenguaje utilicen los códigos de RFC 3066 [\[RFC3066\]](#) o posteriores y para los atributos de "xml:lang" en el caso de documentos basados en XML [\[XML10\]](#). Ver "[FAQ: Two-letter or three-letter language codes.](#)"

Las siguientes reglas determinan las comillas para un documento HTML que está en francés o en alemán:

```
html:lang(fr-ca) { quotes: '« ' ' »' }
html:lang(de) { quotes: '»' '«' '\2039' '\203A' }
:lang(fr) > Q { quotes: '« ' ' »' }
:lang(de) > Q { quotes: '»' '«' '\2039' '\203A' }
```

El segundo par de reglas especifica realmente la propiedad 'quotes' de los elementos Q de acuerdo al lenguaje de su padre. Esto se hace porque la elección del tipo de comillas se basa típicamente en el lenguaje del elemento que rodea la comilla, no en las propias comillas: como este fragmento de francés "à l'improviste" en medio de un texto en inglés usa el tipo de comillas propias del inglés.

5.12 Pseudo-elementos

5.12.1 El pseudo-elemento :first-line

El pseudo-elemento :first-line aplica estilos especiales a la primera línea de un párrafo. Por ejemplo:

```
p:first-line { text-transform: uppercase }
```

La regla anterior significa "convertir las letras de la primera línea de cada párrafo en mayúsculas". Sin embargo, el selector "P:first-line" no equivale a ningún elemento HTML real. Se corresponde con un pseudo-elemento que las [aplicaciones del usuario con conformidad](#) insertará al comienzo de cada párrafo.

Observe que la medida de la primera línea depende de un número de factores, incluyendo el ancho de la página, el tamaño de la fuente, etc. De este modo, un párrafo HTML común como:

```
<P>Este es algún párrafo largo en HTML que será partido en varias líneas.  
La primer línea será identificada por una secuencia ficticia de marcas.  
Las otras líneas serán tratadas como líneas comunes dentro del  
párrafo.</P>
```

las líneas partidas que resultan:

```
ESTE ES ALGÚN PÁRRAFO LARGO EN HTML QUE  
será partido en varias líneas. La primer línea será identificada por una  
secuencia ficticia de marcas.  
Las otras líneas serán tratadas como líneas comunes dentro del párrafo.
```

pueden ser "re-escritas" por las aplicaciones del usuario para incluir la secuencia ficticia de marcas para :first-line. Esta secuencia ficticia de marcas ayuda a mostrar cómo se heredan las propiedades.

```
<P><P:first-line> Este es algún párrafo largo en HTML que</P:first-line>  
será partido en varias líneas. La primer línea será identificada por una  
secuencia ficticia de marcas. Las otras líneas serán tratadas como líneas  
comunes dentro del párrafo.</P>
```

Si un pseudo-elemento divide un elemento real, el efecto deseado a menudo se consigue con una secuencia ficticia de marcas que cierra y reabre el elemento. De este modo, si marcamos el párrafo anterior con un elemento SPAN:

```
<P><SPAN class="test"> Este es algún párrafo largo en HTML que será
partido en varias líneas.</SPAN>La primer línea será identificada por una
secuencia ficticia de marcas. Las otras líneas serán tratadas como líneas
comunes dentro del párrafo.</P>
```

la aplicación del usuario puede generar las marcas de inicio y cierre apropiadas para SPAN cuando inserta la secuencia ficticia de marcas para :first-line.

```
<P><P:first-line><SPAN class="test"> Este es algún párrafo largo en HTML
que</SPAN></P:first-line><SPAN class="test">será partido en varias
líneas.</SPAN> La primer línea será identificada por una secuencia
ficticia de marcas. Las traslíneas serán tratadas como líneas comunes
dentro del párrafo.</P>
```

El pseudo-elemento :first-line sólo puede ser aplicado en elementos a [nivel de bloque](#), de título de tablas (caption) o de celdas de tablas.

El “formato de la primera línea” de un elemento puede ocurrir en el interior de un nivel de bloque (block-level) descendente en el mismo flujo (es decir, i.e., un descendente de nivel de bloque que no ha sido posicionado y no es flotante). Por ejemplo, la primera línea de un DIV en `<DIV><P>Esta línea...</P></DIV>` es la primera línea de P (asumiendo que ambos P y DIV son el nivel de bloque).

La primera línea de una celda o bloque en línea (inline-block) no puede ser la primera línea formateada de un elemento. Así, en `<DIV><P STYLE="display: inline-block">Hola
Adios</P> etcetera</DIV>` la primera línea formateada del DIV no es “Hola”.

Observe que la primera línea de P en este fragmento: `<p>
Primero... no contiene ninguna letra (asumiendo el estilo predeterminado (por defecto) para BR en HTML 4.0). La “primera” palabra no es “Primero” en el formato de la primera línea.`

Una AU debería actuar como si la etiqueta ficticia del principio del pseudo-elemento fist-line estuviese justamente dentro del elemento de nivel de bloque de inclusión. (Ya que CSS1 y CSS2 no dicen nada en este caso, los autores no deberían confiar en este comportamiento) Aquí está el ejemplo. Una secuencia de etiqueta ficticia para:

```
<DIV>
  <P>Primer párrafo</P>
  <P>Segundo párrafo</P>
</DIV>
```

es

```
<DIV>
  <P><DIV:first-line><P:first-line>Primer párrafo </P:first-
line></DIV:first-line></P>
  <P><P:first-line>segundo párrafo </P:first-line></P>
</DIV>
```

El pseudo-elemento `:first-line` es similar a un elemento a nivel de línea, pero con ciertas restricciones. Sólo las siguientes propiedades se aplican al pseudo-elemento `:first-line`: [propiedades de fuentes](#), [propiedades de color](#), [propiedades del fondo](#), 'espacio entre palabras' (`word-spacing`), 'espacio entre letras' (`letter-spacing`), 'decoración del texto' (`text-decoration`), 'alineación vertical' (`vertical-align`), 'transformaciones del texto' (`text-transform`), 'ancho de línea' (`line-height`),. Si bien las AUs (Aplicaciones de Usuarios) pueden aplicarlos a otras propiedades.

5.12.2 El pseudo-elemento `:first-letter`

El pseudo-elemento `:first-letter` puede ser usado para las "capitulares" y "capitulares caídas", que son efectos tipográficos de uso frecuente. Esta clase de letra inicial es similar a un elemento a nivel de línea cuya propiedad `float` es `none`, de otro modo es similar a un elemento flotante.

Estas son las propiedades que se aplican a los pseudo-elementos `:first-letter` pseudo-elements: [propiedades de fuentes](#), 'decoración del texto' (`text-decoration`), 'transformaciones del texto' (`text-transform`), 'espacio entre letras' (`letter-spacing`), 'espacio entre palabras' (`word-spacing`) (cuando sea apropiado), 'ancho de línea' (`line-height`), 'flotante' (`float`), 'alineación vertical' (`vertical-align`) (solo si `float` es `none`), [propiedades de los márgenes](#) [propiedades de relleno](#), [propiedades de los bordes](#), [propiedades de color](#) [propiedades del fondo](#). Si bien las AUs (Aplicaciones de Usuarios) pueden aplicarlos a otras propiedades. Las AUs (Aplicaciones de Usuarios) deben permitir procesar a una topología correcta las letras "capitulares" o "capitulares caída", la AU debe elegir entre el ancho de líneas (`line-height`), con y la formación del ancho base de la letra, si no le gusta para elementos normales. CSS3 tiene expectativas de tener unas propiedades específicas para aplicarlas a `first-letter`.

Este ejemplo muestra una posible interpretación de una "capitular". Observe que el espacio entre líneas (`line-height`) que es fijado por el pseudo-elemento `first-letter` es 1.1, pero la AU (Aplicación del Usuario) en este ejemplo ha computado el ancho de la primera letra de forma diferente, así que no es necesario establecer un espacio entre la primera y segunda línea. Observe también que la etiqueta ficticia del principio está dentro de `SPAN`, y así el grosor de la fuente (`font-weight`) de la primera letra es normal, no negrita como el `SPAN`

```
p { line-height: 1.1 }
p:first-letter { font-size: 3em; font-weight: normal }
span { font-weight: bold }
...
<p><span>Het hemelsche</span> gerecht heeft zich ten lange lesten<br>
Erbarremt over my en mijn benaeuwde vesten<br>
En arme burgery, en op mijn volcx gebed<br>
En dagelix geschrey de bange stad ontzet.
```

Het hemelsche gerecht heeft zich ten lange lesten
Erbarremt over my en mijn benaeuwde vesten
En arme burgery, en op mijn volcx gebed
En dagelix geschrey de bange stad ontzet.

La siguiente CSS2.1 hará una capitular caída que abarca dos líneas:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

<HTML>
  <HEAD>
    <TITLE>Capitular caída</TITLE>
    <STYLE type="text/css">
      P      { font-size: 12pt; line-height: 12pt }
      P:first-letter {font-size: 200%; font-style: italic; font-weight: bold; float: left}
      SPAN    { text-transform: uppercase }
    </STYLE>
  </HEAD>
  <BODY>
    <P><SPAN>Algunas</SPAN> palabras de un artículo en un periódico.</P>
  </BODY>
</HTML>
```

Este ejemplo sería formateado como sigue:

*A*LGUNAS palabras
de un artículo en un
periódico. D

La secuencia ficticia de marcas es:

```
<P>
<SPAN>
<P:first-letter>
A
</P:first-letter>algunas
</SPAN>
palabras de un artículo en un periódico.
</P>
```

Observe que la marca para el pseudo-elemento `:first-letter` es contigua al contenido (es decir, el carácter inicial), mientras que la marca de inicio del pseudo-elemento `:first-line` se inserta después de la marca de inicio del elemento al que está asignado.

Con el propósito de conseguir el formato de las capitulares caídas tradicionales, las aplicaciones del usuario pueden aproximar los tamaños de la fuente, por ejemplo para emparejar las líneas base. Además, el contorno del signo puede ser tenido en cuenta al establecer el formato. La puntuación (es decir, los caracteres definidos en Unicode [UNICODE] en las clases de puntuación "open" (Ps), "close" (Pe), "inicial" (Pi), "final" (Pf) y "other" (Po)), que precede a la primera letra debe ser incluida, como en:

“*M*ás vale pájaro
en mano que
cien volando”,
dice un proverbio. D

`:first-letter` así se aplica si la primera letra es en efecto un número, Ej.: el “6” en “67 millones de dólares es mucho dinero”.

Los pseudo-elementos `:first-letter` se aplican a bloques, listas de objetos (`list-item`), celdas de tablas (`table-cell`), títulos de tablas (`table-caption`) y en elementos de bloques en línea (`inline-block`).

Los pseudos-elementos `:first-letter` pueden ser utilizados como la mayoría de los elementos que contienen texto o que tienen descendentes en el mismo flujo que contiene el texto. Una AU puede actuar como si la etiqueta ficticia de inicio del pseudo-elemento `first-letter` fuese justamente después del primer elemento del texto, aunque este primer texto esté en un descendente.

La etiqueta ficticia de secuencia para este fragmento de HTML:

```
<div>
<p>The first text.
```

es:

```
<div>
<p><div:first-letter><p:first-letter>T</...></...>he first text.
```

La primera letra de una celda de una tabla o de un bloque en línea no puede ser la primera letra de un elemento. así `<DIV><P STYLE="display: inline-block">Hola
Adios</P> etcetera</DIV>` la primera letra del DIV no es la letra "H". De hecho, el DIV no tiene primera letra.

La primera letra debe ocurrir sobre el formato de la primera línea. Por ejemplo, en este fragmento `<p>
Primero...` la primera línea no contiene alguna letra y `:first-letter` no es equivalente a nada (asumiendo el estilo por defecto para los `
` en HTML 4.0). En particular, no se aplica a la "P" de "Primero"

Si un elemento es una [lista de elementos](#) (`display: list-item`), `:first-letter` se aplica a la primera letra en la caja principal después del marcador. Las AUs pueden ignorar `:first-letter` sobre listas de elementos con `list-style-position: inside`. Si un elemento tiene el contenido `:before` o `:after`, `:first-letter` se aplica a la primera letra del elemento incluido en ese contenido.

Ej., después de la regla `'p:before {content: "Note: "}`, el selector `'p:first-letter'` se aplica a la "N" de "Note".

Algunos lenguajes pueden tener reglas específicas acerca de cómo tratar ciertas combinaciones de letras. En alemán, por ejemplo, si la combinación de las letras "ij" aparece al comienzo de una palabra, ambas letras deberían ser consideradas dentro del pseudo-elemento `:first-letter`.

Si la letra que forma la primera letra no está en el mismo elemento, tal como "T" en `<p>' T. . .`, la AU (Aplicación del Usuario) puede crear un pseudos-elemento `first-letter` para uno de los elementos, para ambos elementos o simplemente no crearlo.

El siguiente ejemplo ilustra cómo pueden interactuar los pseudo-elementos que se superponen. La primera letra de cada elemento P será verde con una tamaño de fuente de '24pt'. El resto de la primera línea formateada será azul mientras que el resto del párrafo será rojo.

```
p { color: red; font-size: 12pt }
p:first-letter { color: green; font-size: 200% }
p:first-line { color: blue }
```

<P>Un texto corto que ocupa dos líneas</P>

Asumiendo que el corte de la línea ocurre antes de la palabra "ocupa", la secuencia ficticia de marcas para este fragmento podría ser:

```
<P>
<P:first-line>
<P:first-letter>
S
</P:first-letter>n texto corto que
</P:first-line>
Ocupa dos líneas
</P>
```

Observe que el elemento `:first-letter` está dentro del elemento `:first-line`. Las propiedades adjudicadas a `:first-line` son heredadas por `:first-letter`, pero son anuladas si la misma propiedad es determinada en `:first-letter`.

5.12.3 Los pseudo-elementos `:before` y `:after`

Los pseudo-elementos `:before` y `:after` pueden ser usados para insertar un contenido generado antes o después del contenido del elemento. Son explicados en la sección acerca de [texto generado](#).

```
h1:before {content: counter(chapno, upper-roman) ". " }
```

Cuando los pseudo-elementos `:first-letter` y `:first-line` son combinados con `:before` y `:after`, se aplican a la primera letra o línea del elemento que incluye el texto insertado.

```
p.special:before {content: "Especial! "}
p.special:first-letter {color: #ffd800}
```

Esto procesará la "E" de "Especial!" en un amarillo oro.

6 Asignación de valores a las propiedades, Cascada, y Herencia

Contenido

- [6.1 Valores especificados, computados, y reales](#)
 - [6.1.1 Valores especificados](#)
 - [6.1.2 Valores computados](#)
 - [6.1.3 Valores usados](#)
 - [6.1.4 Valores reales](#)
- [6.2 Herencia](#)
 - [6.2.1 El valor 'inherit'](#)
- [6.3 La regla @import](#)
- [6.4 Cascada](#)
 - [6.4.1 Orden de conexión en cascada](#)
 - [6.4.2 Las reglas !important](#)
 - [6.4.3 Cálculo de especificidad de un selector](#)
 - [6.4.4 Precedencia de las indicaciones de presentación fuera de CSS](#)

6.1 Valores especificados, computados y reales

Una vez que una aplicación del usuario haya analizado un documento y haya construido una [estructura del documento](#), debe asignar, para cada elemento de la estructura, un valor por cada propiedad que es aplicada a los [tipos de medios](#). a los cuales se dirige.

El valor final de una propiedad es el resultado de un cálculo en 4 pasos: el valor se determina por medio de la especificación ("**valor especificado**"), luego resuelto a un valor que se utilice para la herencia ("**valor computado**"), después se convierte en un valor absoluto en caso de necesidad ("**valor usado**"), y finalmente se transforma según las limitaciones del entorno local ("**valor real**").

6.1.1 Valores especificados

Las aplicaciones del usuario deben primero asignar un valor especificado a cada propiedad basados en los mecanismos siguientes (en orden de precedencia):

1. Si la [cascada](#) da lugar a un valor, utilícelo.
2. De lo contrario, si la propiedad es [heredada](#) y el elemento no es el raíz de la estructura del documento , utilice el valor computado del elemento padre.
3. Si no, utilice el valor inicial de la propiedad . El valor inicial de cada propiedad se indica en la definición de la propiedad.

6.1.2 Valores computados

Los valores especificados se resuelven a los valores computados durante la cascada; por ejemplo URIs relativas válidas se hacen absolutas y las unidades 'em' y 'ex' se computan a píxel o a medidas absolutas. Computar un valor nunca requiere que la aplicación del usuario interprete el documento.

El valor computado de URIs no válidas y absolutas es el valor especificado.

Cuando el valor especificado no es 'heredado', el valor computado de una propiedad se determina según lo especificado por la línea del Valor Computado en la definición de la propiedad. Vea la sección sobre [herencia](#) para la definición de valores computados cuando el valor especificado es 'heredado'.

El valor computado existe incluso cuando la propiedad no se aplica, según lo definido por '[se aplica a](#)' la línea. Sin embargo, algunas propiedades pueden definir el valor computado de una propiedad para un elemento que dependa o no de la propiedad aplicada a ese elemento.

6.1.3 Valores usados

Los valores computados pueden ser relativos a otros; por ejemplo una anchura se podría fijar como porcentaje, el cual depende de la anchura del bloque que lo contiene. *El valor usado* es el resultado de tomar el valor computado y de resolver estas dependencias en un valor absoluto final usado para la disposición real.

6.1.4 Valores reales

Un valor usado es en principio el valor usado para procesarlo, pero una aplicación del usuario puede no estar en condiciones de hacer uso del valor en el ámbito determinado. Por ejemplo, una aplicación del usuario sólo puede procesar los bordes con un ancho expresado en píxeles enteros y por consiguiente debe aproximar el ancho computado o la aplicación del usuario puede forzar a utilizar cortinillas solamente en negras o blancas en lugar de todos los colores. El valor real es el valor usado después de que se haya aplicado cualquier aproximación.

6.2 Herencia

Algunos valores son heredados por los hijos de un elemento en la [estructura del documento](#) descrito [arriba](#). Cada propiedad [define](#) si se heredada o no.

Suponga que hay un elemento H1 con un elemento enfatizado (EM) en su interior:

```
<H1>El título < EM>es</EM > importante!</H1 >
```

Si no se ha asignado ningún color al elemento EM, el énfasis en "es" heredará el color del elemento padre, así que si H1 tiene el color azul, el elemento EM estará igualmente en azul.

Cuando ocurre la herencia, los elementos heredan valores computados. El valor computado del elemento padre se convierte, a la vez, en el valor especificado y en el valor computado en el hijo.

Por ejemplo, dado la hoja siguiente del estilo:

```
body { font-size: 10pt }
h1 { font-size: 130% }
```

y este fragmento del documento:

```
<BODY>
  <H1>Un <EM>largo</EM> encabezado</H1>
</BODY>
```

la propiedad 'font-size' del elemento H1 tendrá el valor computado '13pt' (el 130% de 10pt, el valor del padre). Como el valor computado de 'font-size' es heredado, el elemento EM tendrá el valor computado '13pt' también. Si la aplicación del usuario no tiene la fuente de 13pt disponible, el valor real de 'font-size' para H1 y también para EM podría ser, por ejemplo, '12pt'.

6.2.1 El valor 'inherit' (heredado)

Cada propiedad también puede tener un valor especificado como 'inherit' (heredado), lo cual significa que, para un elemento determinado, la propiedad toma el mismo [valor computado](#) que la propiedad del elemento padre. El valor 'inherit' puede ser utilizado para reforzar los valores heredados, y también puede ser utilizado sobre propiedades que normalmente no se heredan.

En el ejemplo de abajo, las propiedades 'color' y 'background' son definidas para el elemento BODY. En todos los demás elementos, el valor de 'color' será heredado y el fondo será transparente. Si estas reglas son parte de la hoja de estilo del usuario, el texto negro sobre un fondo blanco será forzado en todo el documento.

```
body {
  color: black !important;
  background: white !important;
}

* {
  color: inherit !important;
  background: transparent !important;
}
```

6.3 La regla @import

La regla '@import' permite a los usuarios importar hojas de estilo desde otras hojas de estilo. Cualquier regla @import debe preceder a todas las reglas especificadas en una hoja de estilo. La palabra clave '@import' debe ir seguida por el URI de la hoja de estilo a incluir. También se permite una cadena; que será interpretada como si contuviera url(...) en torno a ella.

Las líneas siguientes son equivalentes en su significado e ilustran ambas la sintaxis de '@import' (uno con "url()") y la otra con solo una cadena):

```
@import "mystyle.css";
@import url("mystyle.css");
```

Para que las aplicaciones del usuario puedan evitar recuperar recursos para los [tipos de medios](#), no soportados, los autores pueden especificar reglas @import dependientes de los medios. Estas importaciones condicionales especifican los tipos de medios, separados por comas, después del URI.

Las reglas siguientes ilustran cómo las reglas del @import se pueden hacer medio-dependientes:

```
@import url("fineprint.css") print;  
@import url("bluish.css") projection, tv;
```

En ausencia de cualquier tipo de medios, la importación es incondicional. La especificación de 'all' para el medio tiene el mismo efecto.

6.4 Cascada

Las hojas de estilo pueden tener tres orígenes diferentes: el autor, el usuario y la aplicación del usuario.

- **Autor** . El autor especifica las hojas de estilo para un documento fuente según las convenciones del lenguaje del documento. Por ejemplo, en HTML, las hojas de estilo se pueden incluir en el documento o vincularlas externamente.
- **Usuario** : El usuario puede especificar la información de estilo para un documento particular. Por ejemplo, el usuario puede especificar un archivo que contenga una hoja de estilo o la aplicación del usuario puede proporcionar una interfaz que genere una hoja de estilo del usuario (o se comporta como si lo hiciera).
- **Aplicaciones del usuario:** [Las aplicaciones del usuario con conformidad](#) deben aplicar *una hoja de estilo predeterminada* (o compórtese como si lo hicieran) previa a todas las hojas de estilo para un documento. La hoja de estilo predeterminada de una aplicación del usuario debe presentar los elementos del lenguaje del documento de un modo que satisfaga las expectativas generales de la presentación del lenguaje del documento (Ej., para los navegadores visuales, el elemento EM en el HTML se presenta usando una fuente itálica). Vea "[Un ejemplo de hoja de estilo para HTML](#)" para una hoja de estilo predeterminada que se recomienda para los documentos HTML 4.0.

Observe que el usuario puede modificar los ajustes del sistema (Ej.. colores del sistema) que afectan a la hoja de estilo predeterminada (por defecto). Sin embargo, en la práctica nos encontramos con aplicaciones de usuarios que hacen imposible cambiar los valores en la hoja de estilo predeterminada.

Las hojas de estilo con estos tres orígenes se superpondrán en su acción e interactuarán de acuerdo a la cascada.

La cascada de CSS asigna un peso (importancia) a cada regla del estilo. Cuando varias reglas se aplican, la de mayor peso toma mayor preponderancia (precedencia)

Por defecto, las reglas en las hojas de estilo del autor tienen más peso que las reglas en las hojas de estilo del usuario. La preponderancia (precedencia) se invierte, sin embargo, con la regla "!important". Todas las reglas del usuario y del autor tienen más peso que las reglas de la hoja de estilo predeterminada en la AU (Aplicación del Usuario).

6.4.1 Orden de cascada

Para encontrar el valor para una combinación de elemento/propiedad, las aplicaciones del usuario deben aplicar la siguiente orden de disposición:

1. Encontrar todas las declaraciones que se aplican al elemento y a la propiedad en cuestión, para el [tipo de medio](#) al que está dirigido. Las declaraciones se aplican si el selector asociado [coincide](#) con el elemento en cuestión.
2. Clasificar por la importancia (normal o importante) y por el origen (autor, usuario, o aplicación del usuario). En orden ascendente:
 1. hojas de estilo de la aplicación del usuario
 2. hojas de estilo normales del usuario
 3. hojas de estilo normales del autor
 4. hojas de estilo importantes del autor
 5. hojas de estilo importantes del usuario
3. Clasificar por la [especificidad](#) del selector: Los selectores más específicos sustituirán a los más generales. Los Pseudo-elementos y las pseudo-clases se cuentan como elementos y clases normales, respectivamente.
4. Finalmente, clasificar por el orden especificado: si dos reglas tienen el mismo peso, origen y especificidad, la última en ser especificada vence. Las reglas en hojas de estilo importadas se consideran que están antes de cualquier regla en la propia hoja de estilo.

Aparte de la definición de "!important" en declaraciones individuales, esta estrategia da a las hojas de estilo del autor un peso más relevante que los del lector. Es por lo tanto importante que la aplicación del usuario le brinde al usuario la posibilidad de neutralizar la influencia de determinada hoja de estilo, Ej., a través de un menú desplegable.

6.4.2 Las reglas !important

CSS procura crear un equilibrio de poder entre el autor y el usuario de las hojas de estilo. Por defecto, las reglas de la hoja de estilo del autor sustituyen las de la hoja de estilo del usuario (véase la regla 3 de cascada).

Sin embargo, para el equilibrio, una declaración "!important" (el símbolo delimitador "!" y la palabra clave "important" siguiendo a la declaración) toma precedencia sobre una declaración normal. Las hojas de estilo del autor y del usuario pueden contener declaraciones "!important", y las reglas "!important" del usuario sustituyen a las reglas "!important" del autor. Esta propiedad del CSS mejora la accesibilidad de documentos dándole a los usuarios con requisitos especiales (fuentes grandes, combinaciones de color, etc.) el control sobre la presentación.

Declarar una propiedad resumida (Ej., ['background'](#)) como "!important" es equivalente a declarar todas sus sub-propiedades como "!important".

La primer regla en la hoja de estilo del usuario en el ejemplo siguiente contiene una declaración "!important", que sustituye a la correspondiente declaración en la hoja de estilo del autor. La segunda declaración también resultará vencedora debido a que a sido marcada como "!important". Sin embargo, la tercera regla en la hoja de estilo del usuario no es "!important" y entonces perderá con la segunda regla en la hoja de estilo del autor (que definió el estilo en una propiedad resumida). Asimismo, la tercera regla del autor perderá con la segunda regla del

autor debido a que la segunda regla es "important". Esto muestra que las declaraciones "!important" tienen la misma función también dentro de las hojas de estilo del autor.

```
/* De la hoja de estilo del usuario */
P { text-indent: 1em ! important }
P { font-style: italic ! important }
P { font-size: 18pt }/* De la hoja de estilo del autor */
P { text-indent: 1.5em !important }
P { font: 12pt sans-serif !important }
P { font-size: 24pt }
```

6.4.3 Cálculo de la especificidad de un selector

La especificidad de un selector se calcula como sigue:

- Se cuenta 1 si el selector es más bien un atributo de "estilo" que un selector, 0 en caso contrario (=a) (en HTML, los valores de un atributo "estilo" de un elemento son reglas de hojas de estilo. Estas reglas no tienen selectores, así a=1, b=0, c=0 y d=0)
- Se cuenta el número de atributos ID en el selector (= b)
- Se cuenta el número de los atributos y pseudo-clases en el selector (= c)
- Se cuenta el número de nombres de elementos y pseudos-elementos en el selector (= d)

La especificidad está basada solamente sobre la forma de los selectores. En particular, un selector de la forma "[id=p33]" es contado como un selector de atributo (a=0, b=0, c=1, d=0), aunque el atributo sea definido como un "ID" en el código DTD del documento

La concatenación de los cuatro números a-b-c-d (en un sistema numérico de base mayor) proporciona la especificidad.

Algunos ejemplos:

```
*          {} /* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li         {} /* a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li      {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li   {} /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up]{} /* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red {} /* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level {} /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y      {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style=""   {} /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */

<HEAD>
<STYLE type="text/css">
  #x97z { color: red }
</STYLE>
</HEAD>
<BODY>
<P ID=x97z style="color: green">
</BODY>
```


En el ejemplo de arriba, el color del elemento P sería verde. La declaración en el “style” sustituye a la del elemento STYLE debido a la regla 3 de cascada, ya que tiene una especificidad mayor.

6.4.4 Precedencia de las indicaciones de presentación fuera de CSS

Las AU pueden elegir cumplir con los atributos de presentación en un código fuente de un documento HTML. Si es así, estos atributos son trasladados a las reglas CSS correspondientes con la especificidad igual a cero y son tratadas como si fueran insertadas al principio de la hoja de estilo del autor. Estas pueden ser sustituidas por reglas subsecuentes de la hoja de estilo en la fase de transición, esta política lo hará más fácil para que coexistan atributos estilísticos con hojas de estilo

Para HTML, cualquier atributo que no está en la lista siguiente debería ser considerado presentacional: abbr, accept-charset, accept, accesskey, action, alt, archive, axis, charset, checked, cite, class, classid, code, codebase, codetype, colspan, coords, data, datetime, declare, defer, dir, disabled, enctype, for, headers, href, hreflang, http-equiv, id, ismap, label, lang, language, longdesc, maxlength, media, method, multiple, name, nohref, object, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onload, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onreset, onselect, onsubmit, onunload, onunload, profile, prompt, readonly, rel, rev, rowspan, scheme, scope, selected, shape, span, src, standby, start, style, summary, title, type (excepto sobre los elementos LI, OL y UL), usemap, value, valuetype, version.

Para otros lenguajes, toda la base del lenguaje de estilo debería ser manejada en la hoja de estilo de la aplicación del usuario.

El usuario siguiente de hoja de estilo anularía el font-weight del elemento 'b' en todos los documentos, y el color del elemento 'font' con atributos de color en documentos XML. Esto no afectaría al color de ningún elemento 'font' con el atributo color en documentos HTML:

```
b { font-weight: normal; }
font[color] { color: orange; }
```

El siguiente, sin embargo, anularían el color de elementos fuente (font) en todos los documentos:

```
font[color] { color: orange ! important; }
```

7 tipos de medios

Contenido

- [7.1 Introducción a los tipos de medios](#)
- [7.2 Especificar las hojas de estilo dependientes de los medios](#)
 - [7.2.1 La regla del @media](#)
- [7.3 Tipos de medios reconocidos](#)
 - [7.3.1 Grupos de medios](#)

7.1 Introducción a los tipos de medios

Una de las características más importantes de las hojas de estilo es que especifican cómo debe ser presentado un documento en diferentes medios: en la pantalla, en el papel, con un sintetizador de voz, con un dispositivo braille, etc.

Ciertas propiedades del CSS se diseñan solamente para ciertos medios (Ej.. la propiedad `'page-break-before'` se aplica solamente a los medios paginados). En ocasiones, sin embargo, las hojas de estilo para diferentes tipos de medios pueden compartir una propiedad, pero requieren de valores diferentes para esa propiedad. Por ejemplo, la propiedad `'font-size'` es útil para los medios de pantalla y de impresión. Estos dos tipos de medios son lo suficientemente diferentes como para requerir diversos valores para la propiedad común; un documento necesitará típicamente una fuente más grande en una pantalla de ordenador que sobre el papel. Por lo tanto, es necesario expresar que una hoja de estilo, o una sección de una hoja de estilo, se aplica a ciertos tipos de medios.

7.2 Especificación de hojas de estilo dependientes de los medios

Hay actualmente dos maneras de especificar las dependencias de los medios para las hojas de estilo:

- Especificar el medio de destino desde una hoja de estilo con las reglas arroba: `@media` o `@import`.

```
@import url("fancyfonts.css") screen;
@media print {
  /* la hoja de estilo para impresión va aquí */
}
```

- Especificando el medio de destino dentro del lenguaje del documento. Por ejemplo, en HTML 4,0 ([\[HTML40\]](#)), el atributo "media" en el elemento LINK especifica los medios destino de una hoja de estilo externa:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vínculo a un medio</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      MEDIA="print, handheld" HREF="foo.css">
  </HEAD>
  <BODY>
    <P>The body...
  </BODY>
</HTML>
```

La regla [@import](#) se define en el [capítulo sobre cascada](#).

7.2.1 La regla @media

Una regla @media especifica los [tipos de medios](#) de destino (separados por comas) de un sistema de reglas (delimitadas por llaves). La construcción de @media le permite a las hojas de estilo reglas para distintos medios en la misma hoja de estilo:

```
@media print {
  body { font-size: 10pt }
}
@media screen {
  body { font-size: 13px }
}
@media screen, print {
  body { line-height: 1.2 }
}
```

7.3 Tipos de medios reconocidos

Los nombres elegidos para los tipos de medios de CSS reflejan los dispositivos de destino para los cuales las propiedades relevantes tienen sentido. En el siguiente listado de tipos de medios de CSS, las descripciones entre paréntesis no son normativas. Solamente brindan un sentido de la clase de dispositivo del tipo de medio a que se hace referencia.

all

Conveniente para todos los dispositivos.

braille

Previsto para los dispositivos táctiles de braille.

embossed

Previsto para las impresoras de paginas braille.

handheld

Previsto para los dispositivos de mano (pantalla típicamente pequeña, ancho de banda limitada).

print

Previsto para el material paginado y para los documentos percibidos en pantalla en modo de vista previa a la impresión. Por favor, consulte la sección sobre [medios paginados](#) para información sobre los formatos de publicación que son específicos de los medios paginados.

projection

Previsto para las presentaciones proyectadas, por ejemplo proyectores. Por favor, consulte la sección sobre [medios paginados](#) para información sobre los formatos de publicación que son específicos de los medios paginados.

screen

Previsto sobre todo para las pantallas de ordenadores a color.

speech

Previsto para los sintetizadores de voz. Nota: CSS2 tenía un tipo de medios similar llamado 'auditivos' para este propósito. Vea el apéndice [hojas de estilo auditivas](#) para los detalles.

tty

Previsto para los medios que usan una rejilla de caracteres de espacio fijo (tales como teletipos, terminales, o dispositivos portátiles con limitada capacidad del monitor). Los autores no deben utilizar [unidades en píxel](#) con el tipo de medio "tty".

TV

Previsto para los dispositivos de tipo televisión (resolución baja, color, pantallas con limitado desplazamiento, sonido disponible).

Los nombres de los tipos de medios no diferencian entre mayúsculas y minúsculas (case-insensitive).

Los tipos de medios son mutuamente excluyentes en el sentido de que una aplicación del usuario puede soportar solamente un tipo de medio al interpretar un documento. Sin embargo, las aplicaciones del usuario pueden tener diversos modos que soporten diferentes tipos de medios.

Los nombres desconocidos de tipo de medio no deben dar lugar a la regla @media que es ignorada.

Nota. Las versiones futuras de CSS pueden extender la lista de los tipos de medios. Los autores no deben contar con nombres de tipos de medios que todavía no son definidos por una especificación CSS.

7.3.1 Grupos de medios

Esta sección es informativa, no normativa.

Cada definición de propiedad en CSS especifica a que tipos de medio se aplica. Puesto que las propiedades se aplican generalmente a varios tipos de medios, la sección "Se aplica a los medios" de cada definición de propiedad lista los grupos de medios en vez de los tipos de medio individuales. Cada propiedad se aplica a todos los tipos de medios en los grupos de medios listados en su definición.

El CSS 2.1 define los grupos siguientes de medios:

- **Continuos (continuous) o paginados (paged) .**
- **Visual (visual), audio (audio), voz (speech), o táctil (tactile) .**
- **Rejilla (gris) (para dispositivos de rejilla de caracteres), o mapa de bit (bitmap) .**
- **Interactivos (interactive) (para los dispositivos que permiten la interacción con el usuario), o estáticos (static) (para los que no lo hacen).**
- **Todos (all) (incluye todos los tipos de medios)**

La tabla siguiente muestra las relaciones entre los grupos de medios y los tipos de medios:

Relación entre los grupos de medios y los tipos de medios

Tipos De Medios	Grupos De Medios			
	continuos/ paginados	visual/audio/ voz/táctiles	rejillas/bitmap	interactivos/ estáticos
braille	continuo	táctil	rejilla	ambos
embossed	paginado	táctil	rejilla	estático
handheld	ambos	visual, audio, voz	ambos	ambos
print	paginado	visual	BITMAP	estático
projection	paginado	visual	BITMAP	interactivo
Screen	continuo	visual, audio	BITMAP	Ambos
Speech	continuo	Voz	N/A	Ambos
Tty	continuo	visual	rejilla	Ambos
TV	ambos	visual, audio	BITMAP	Ambos

8 Modelo de cajas

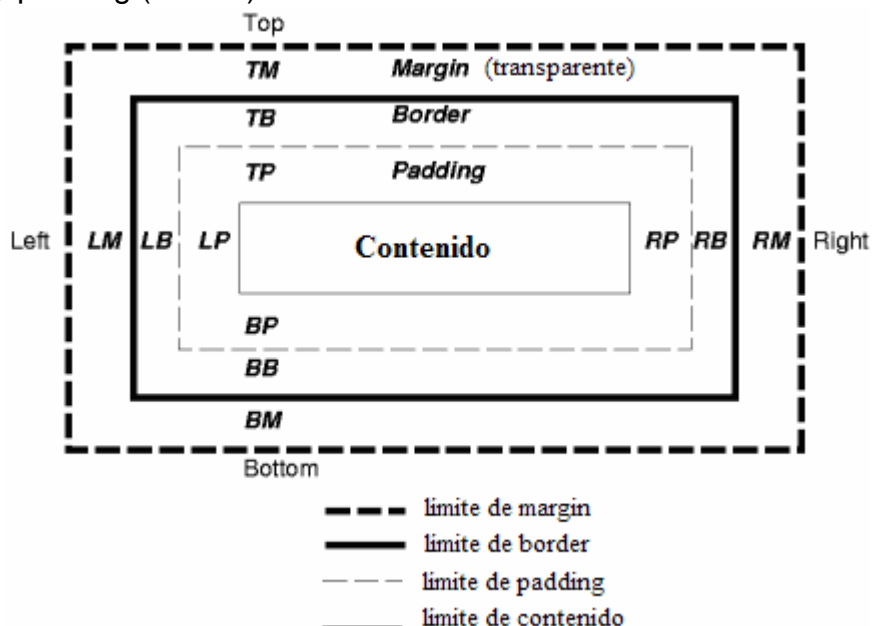
Contenido

- [8.1 Dimensiones de la caja](#)
- [8.2 Ejemplo de márgenes, rellenos y bordes](#)
- [8.3 propiedades del margen: 'margin-top' 'margin-right' 'margin-bottom' 'margin-left' y 'margin'](#)
 - [8.3.1 Márgenes cerrados](#)
- [8.4 Propiedades de relleno: 'padding-top' 'padding-right' 'padding-bottom' 'padding-left' y 'padding'](#)
- [8.5 Propiedades del borde](#)
 - [8.5.1 Ancho del borde: 'border-top-width' 'border-right-width' 'border-bottom-width' 'border-left-width' y 'border-width'](#)
 - [8.5.2 Color del borde: 'border-top-color' 'border-right-color' 'border-bottom-color' 'border-left-color' y 'border-color'](#)
 - [8.5.3 Estilo del borde: 'border-top-style' 'border-right-style' 'border-bottom-style' 'border-left-style' y 'border-style'](#)
 - [8.5.4 propiedades resumidas del borde: 'border-top' 'border-left' 'border-bottom' 'border-right' y 'border'](#)
- [8.6 El modelo de caja para los elementos en línea en contexto bidireccional](#)

El modelo de caja de CSS describe las cajas rectangulares que se generan para los elementos en la [estructura del documento](#) y se presentan según el [modelo de formato visual](#).

8.1 Dimensiones de la caja

Cada caja tiene *un área de contenido* (Ej., texto, una imagen, etc.) y las áreas circundantes opcionales de padding (relleno), border (borde) y margin (margen); el tamaño de cada área es especificado por las propiedades definidas abajo. El diagrama siguiente muestra cómo se relacionan estas áreas y la terminología usada para referirse a las partes de margin (margen), border (borde) y padding (relleno):



El margen (margin), el borde (border), y el relleno (padding) pueden ser divididos en los segmentos superior (top), derecho (right), inferior (bottom) e izquierdo (left) (Ej., en el diagrama, "LM" para el margen izquierdo (Left Margin), "RP" para el relleno derecho (Right Padding), "TB" para el borde superior (Top Border), etc.).

El perímetro de cada una de las cuatro áreas (contenido, padding, border y margin) es llamado "límite", de manera que cada caja tiene cuatro límites:

Límite del contenido o límite interno

El límite del contenido rodea el rectángulo dado por [la anchura](#) (width) y [la altura](#) (height) de la caja, que dependen a menudo del [contenido procesado](#) del elemento.

Límite de padding (relleno)

El límite de padding (relleno) rodea a la caja de relleno (padding). Si padding tiene 0 de anchura, el límite del relleno (padding) es el mismo que el límite de Contenido.

Límite de border (borde)

El límite de border (borde) rodea el borde de la caja. Si border tiene 0 de anchura, el límite del borde es igual que el límite de padding (relleno).

Límite de margin (margen) o límite externo

El límite de margin (margen) rodea el margen de la caja. Si margin tiene 0 de anchura, el límite del margen es igual al límite del borde (border).

Cada límite se puede dividir en límite superior (top), derecho (right), inferior (bottom) e izquierdo (left).

Las dimensiones del área de contenido de una caja -- *la anchura del contenido* y *la altura del contenido* -- dependen de varios factores: si el elemento que genera la caja tiene asignada las propiedades '[width](#)' (anchura) o '[height](#)' (altura), si la caja contiene texto u otras cajas, si la caja es una tabla, etc. La anchura y la altura de la caja se discuten en el capítulo sobre [detalles del modelo de composición visual](#).

El estilo del fondo (background) del contenido, el relleno (padding) y las áreas de los bordes (border) de una caja son especificados por la propiedad '[background](#)' del elemento generado. Los márgenes de los fondos (backgrounds) son siempre transparentes.

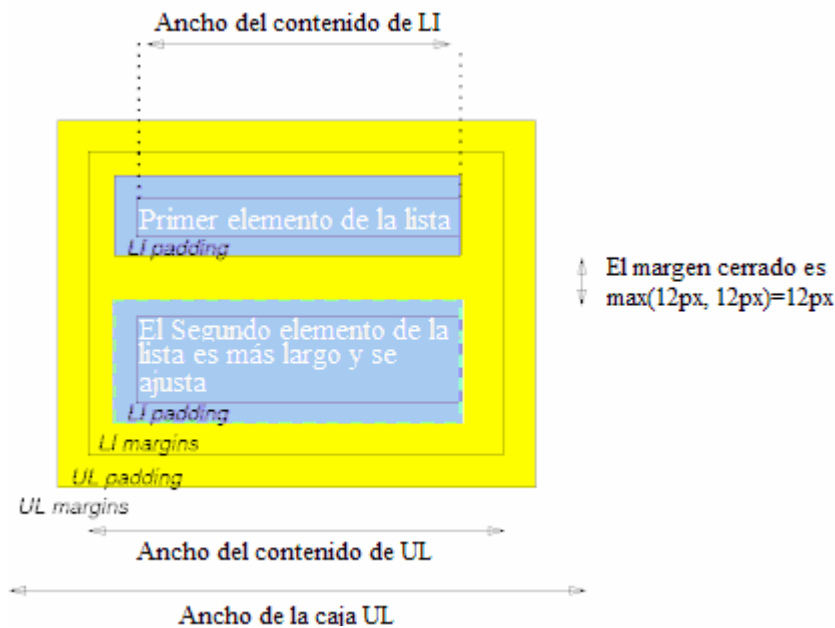
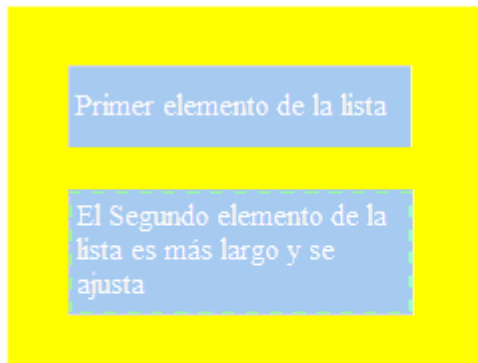
8.2 Ejemplo de márgenes (margin), de rellenos (padding) y bordes (border)

Este ejemplo ilustra cómo interactúan los márgenes, los rellenos y los bordes. El documento HTML del ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Ejemplo de márgenes, rellenos y bordes</TITLE>
    <STYLE type="text/css">
      UL {
        background: green;
        margin: 12px 12px 12px 12px;
        padding: 3px 3px 3px 3px;
                                                    /* No se especifica border */
      }
      LI {
        color: black;                               /* el color del texto es negro */
        background: gray;                           /* Contenido, padding será gris */
        margin: 12px 12px 12px 12px;
        padding: 12px 0px 12px 12px; /* Note 0px para padding right */
        list-style: none                            /* sin viñeta antes de un ítem */
                                                    /* Ningún borde definido */
      }
      LI.withborder {
        border-style: dashed;
        border-width: medium;                       /* pone el ancho de border para todos
los lados */
        border-color: black;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>El primer elemento
<LI class="withborder">El segundo elemento de la lista es
      más largo y se ajusta.
    </UL>
  </BODY>
</HTML>
```

Resulta en una [estructura del documento](#) con (entre otras relaciones) un elemento UL que tiene dos LI hijos.

El primero de los diagramas siguientes ilustra el resultado que produciría este ejemplo. El segundo ilustra la relación entre los márgenes, el relleno y los bordes del elemento UL y la de sus elementos hijos LI.



[D]

Observe que

- El [ancho del contenido](#) para cada caja de LI es calculado de arriba hacia abajo; el [bloque de contención](#) de cada caja LI es establecido por el elemento UL.
- El límite de la altura del margen de cada caja LI depende de la [altura del contenido](#), más el relleno, bordes y márgenes superiores e inferiores. Observe que los márgenes verticales entre las cajas LI se [cierran](#).
- El relleno derecho de las cajas LI se ha fijado a un ancho igual a cero (la propiedad '[padding](#)'). El efecto es evidente en la segunda ilustración.
- Los márgenes de las cajas LI son transparentes — los márgenes son siempre transparentes — por eso el color del fondo (amarillo –yellow) de las áreas de relleno y del contenido de UL se ven a través de ellos.
- El segundo elemento LI especifica un borde rayado (la propiedad '[border-style](#)').

8.3 Propiedades del margen: 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' y 'margin'

Las propiedades del margen especifican la anchura del [área del margen](#) de una caja. La propiedad resumida 'margin' establece el margen para los cuatro lados, mientras que las otras propiedades del margen solamente establecen su lado respectivo. Estas propiedades se aplican a todos los elementos, pero los márgenes verticales no tendrán ningún efecto en los elementos en línea (inline) no reemplazados.

Las propiedades definidas en esta sección se refieren al tipo de valor para **<margin-width>**, que puede tomar uno de los valores siguientes:

< medida >

Especifica una anchura fija.

< porcentaje >

El porcentaje se calcula con respecto a *la anchura* del [bloque de contención](#) de la caja generada. **Observe que también, esto es así para 'margin-top' y 'margin-bottom'.** Si el ancho del bloque de contención depende de este elemento, entonces la disposición que resulta es indefinida en CSS 2.1.

auto

Vea la sección sobre [computando anchos y márgenes](#) para su comportamiento.

Los valores negativos para las propiedades del margen son permitidos, pero puede haber límites específicos de la implementación.

'margin-top' 'margin-bottom'

Valor: [<margin-width>](#) | [inherit](#)

Inicial: 0

Se aplica a: todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'

Se hereda: no

Porcentajes: Referido al ancho del bloque de contención

Medio: [visual](#)

Valor computado: el porcentaje especificado o la medida absoluta

'margin-right' 'margin-left'

Valor: [<margin-width>](#) | [inherit](#)

Inicial: 0

Se aplica a: todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'

Se hereda: no

Porcentajes: Referido al ancho del bloque de contención

Medio: [visual](#)

Valor computado: el porcentaje especificado o la medida absoluta

Estas propiedades asignan el margen superior (top), derecho (right), inferior (bottom) e izquierdo (left) de una caja.

```
h1 { margin-top: 2em }
```

'margin'

Valor:	<code><margin-width> {1,4} inherit</code>
Inicial:	vea las propiedades individuales
Se aplica a:	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'
Se hereda:	no
Porcentajes:	referido al ancho del bloque de contención
Medios:	<u>visual</u>
Valor computado:	vea las propiedades individuales

La propiedad 'margin' es una propiedad resumida para fijar 'margin-top', 'margin-right', 'margin-bottom', y 'margin-left' en un mismo lugar de la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los márgenes superior e inferior son determinados por el primer valor y los márgenes derecho e izquierdo son determinados por el segundo. Si hay tres valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, estos se aplican al superior, derecho, inferior e izquierdo, respectivamente.

```
BODY { margin: 2em } /* todos los márgenes en 2em */
BODY { margin: 1em 2em } /* top & bottom = 1em, right & left = 2em */
BODY { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

La última regla del ejemplo de arriba es equivalente al ejemplo de abajo:

```
BODY {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em; /* copiado del lado opuesto (right) */
}
```

8.3.1 Márgenes cerrados

En esta especificación, la expresión *márgenes cerrados* significa que los márgenes adyacentes (ningún contenido vacío, área de relleno o de borde o espacio libre los separa) de dos o más cajas (que pueden estar una al lado de la otra o anidadas) se combinan para formar un solo margen.

En CSS 2.1, los márgenes horizontales nunca se cierran.

Los márgenes verticales pueden cerrarse entre ciertas cajas:

- Dos o más márgenes verticales adyacentes de cajas de [bloques](#) en el [flujo normal](#) se cierran. El ancho del margen resultante es el máximo de los anchos de los márgenes adyacentes. En el caso de márgenes negativos, el máximo absoluto de los márgenes adyacentes negativos es restado del máximo de los márgenes adyacentes positivos. Si no hay ningún margen positivo, el máximo absoluto de los márgenes adyacentes negativos es cero. **Nota.** Las cajas adyacentes pueden ser generados por elementos que no se relacionan como hermanos o ascendentes.

- Los márgenes verticales entre una caja [flotante](#) y cualquier otra caja no se cierran (ni siquiera entre una flotante y el flujo en su hijo).
- Los márgenes de elementos con '[desbordamiento](#)' con excepción de los 'visibles' no se cierran con sus hijos en el flujo.
- Los márgenes de cajas posicionadas [absolutamente](#) no se cierran (ni siquiera con su hijo en flujo)
- Los elementos de bloques en línea no se cierran (ni siquiera con su hijo en el flujo)
- Si los márgenes superiores (top) e inferiores (bottom) de una caja son adyacentes, entonces es posible para los márgenes cerrarse con ella. En este caso, la posición del elemento depende de su relación con los otros elementos cuyos márgenes están cerrados.
 - Si los márgenes del elemento están cerrados con el margen superior del padre, el límite del borde superior de la caja es definido para ser el mismo que el del padre.
 - Si no, o el padre del elemento no participa en el cierre del margen, o solamente el margen inferior (bottom) del padre está implicado. La posición del límite del borde superior del elemento es la misma que tendría si el elemento tuviera un borde superior no nulo (distinto de cero).

Un elemento que ha tenido aplicado [espacio libre](#) nunca cierra su margen superior con el margen inferior del bloque del padre.

Observe que las posiciones de los elementos que se han cerrado no tienen ningún efecto en las posiciones de los otros elementos cuyos márgenes se están cerrando; la posición del límite del borde superior solamente se requiere para presentar los descendientes de estos elementos.

- Los márgenes de los elementos de la caja raíz no se cierran .

El margen inferior del elemento a nivel de bloque en el flujo es siempre adyacente al margen superior siguiente del hermano a nivel de bloque en el flujo, a menos que ese hermano tenga [espacio libre](#)

El margen superior de un elemento a nivel de bloque en el flujo es adyacente al margen superior del primer hijo a nivel de bloque en el flujo si el elemento no tiene borde superior, relleno superior, y el hijo no tiene [espacio libre](#).

El margen inferior de un elemento a nivel de bloque en el flujo con '[altura](#)' igual a 'auto' y '[altura mínima](#)' (min-height) menor que la altura usada por el elemento es adyacente al margen inferior del último hijo a nivel de bloque en el flujo si el elemento no tiene rellenos inferiores y borde.

Los márgenes propios de un elemento son adyacentes si la propiedad '[altura mínima](#)' (min-height) es cero, y no tiene ni borde vertical, ni relleno vertical, y tiene una '[altura](#)' de cero o 'auto', y no contiene una caja de línea, y todos los márgenes del hijo (si lo hay) en el flujo son adyacentes.

Cuando los propios márgenes de los elementos se cierran, y ese elemento ha tenido espacio libre para aplicarlo, su margen superior se cierra con el margen adyacente de la sub-secuencia hermana pero ese margen resultante no es adyacente con el margen inferior del bloque padre.

El cierre está basado sobre el valor usado en '[padding](#)', '[margin](#)', and '[border](#)' (es decir, después de resolver algunos porcentajes). El margen cerrado es calculado sobre el valor usado sobre varios márgenes.

Por favor consulte los [ejemplos de márgenes, rellenos y bordes](#) para una ilustración de márgenes cerrados.

8.4 Propiedades del relleno: '[padding-top](#)', '[padding-right](#)', '[padding-bottom](#)', '[padding-left](#)', y '[padding](#)'

Las propiedades del relleno especifican el ancho del [área de relleno](#) de una caja. La propiedad resumida '[padding](#)' define el relleno para los cuatro lados mientras que las otras propiedades de relleno sólo definen sus respectivos lados.

Las propiedades definidas en esta sección se refieren al tipo de valor de <padding-width>, que puede tomar uno de los siguientes valores:

<medida>

Especifica un ancho fijo.

<porcentaje>

El porcentaje es calculado con respecto al ancho del [bloque de contención](#) de la caja generada, aún para '[padding-top](#)' y '[padding-bottom](#)'. Si el ancho del bloque de contención depende de este elemento, entonces el resultado mostrado es indefinido en CSS2.1.

A diferencia de las propiedades del margen, los valores del relleno no pueden ser negativos. Al igual que las propiedades del margen, los valores de porcentaje para las propiedades del relleno se refieren al ancho del bloque de contención de la caja generada.

'[padding-top](#)', '[padding-right](#)', '[padding-bottom](#)', '[padding-left](#)'

Valor: <padding-width> | [inherit](#)

Inicial: 0

Se aplica a: todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table' y 'table-cell'

Se hereda: no

Porcentajes: Referido al ancho del bloque de contención

Medio: [visual](#)

Valor Computado: Porcentaje especificado o la medida (longitud) absoluta

Esta propiedad determina el relleno superior (top), derecho (right), inferior (bottom) e izquierdo (left) de una caja.

```
blockquote { padding-top: 0.3em }
```

'padding'

Valor:	<padding-width>{1,4} inherit
Inicial:	Ver propiedades individuales
Se aplica a	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table' y 'table-cell'
Se hereda	no
Porcentajes	Referido al ancho del bloque de contención
Medio	visual
Valor Computado	Ver propiedades individuales

La propiedad 'padding' es una propiedad resumida para definir las propiedades 'padding-top', 'padding-right', 'padding-bottom', y 'padding-left' en un mismo lugar en la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los rellenos superior e inferior son determinados por el primer valor y los rellenos derecho e izquierdo son determinados por el segundo. Si hay tres valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, ellos se aplican al superior, derecho, inferior e izquierdo, respectivamente.

El color de la superficie o la imagen del área de relleno es especificado a través de la propiedad 'background':

```
h1 {  
  background: white;  
  padding: 1em 2em;  
}
```

El ejemplo de arriba especifica un relleno vertical igual a '1em' ('padding-top' y 'padding-bottom') y un relleno horizontal de '2em' ('padding-right' y 'padding-left'). La unidad 'em' es relativa al tamaño de la fuente del elemento: '1em' es igual al tamaño de la fuente en uso.

8.5 Propiedades del Borde

Las propiedades del borde especifican el ancho, color y estilo del [área del borde](#) de una caja. Estas propiedades se aplican a todos los elementos.

Nota. Particularmente para HTML, las aplicaciones del usuario pueden procesar los bordes de ciertos elementos (ej., botones, menús, etc.) de manera diferente a los elementos "ordinarios".

8.5.1 Ancho del borde: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width', y 'border-width'

Las propiedades del ancho del borde especifican el ancho del área del borde. Las propiedades definidas en esta sección se refieren al tipo de valor de <border-width>, que puede tomar uno de los siguientes valores:

thin

Un borde fino.

medium

Un borde mediano.

thick

Un borde grueso.

<medida>

El grosor del borde tiene un valor específico. Las dimensiones del borde explícitas no pueden ser negativas.

La interpretación de los primeros tres valores depende de la aplicación del usuario. Sin embargo, se deben mantener las siguientes relaciones:

'thin' <='medium' <='thick'.

Además, estos grosores deben mantenerse constantes a lo largo de todo un documento.

'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width'

Valor: <border-width> | inherit
Inicial: medium
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medio: visual
Valor Computado: Medida absoluta; '0' si el estilo del borde es 'none' o 'hidden'

Estas propiedades determinan al ancho de los bordes superior (top), derecho (right), inferior (bottom) e izquierdo (left) de una caja.

'border-width'

Valor: <border-width>{1,4} | inherit
Inicia: Ver propiedades individuales
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medio: visual
Valor Computado: Ver propiedades individuales

Esta propiedad es una propiedad resumida para especificar 'border-top-width', 'border-right-width', 'border-bottom-width' y 'border-left-width' en un mismo lugar en la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los bordes superior e inferior son determinados por el primer valor y los bordes derecho e izquierdo son determinados por el segundo. Si hay tres valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, estos se aplican al superior, derecho, inferior e izquierdo, respectivamente.

En los ejemplos de abajo, los comentarios indican los grosores resultantes en los bordes superior, derecho, inferior e izquierdo:

```
h1 { border-width: thin } /* thin thin thin thin */
h1 { border-width: thin thick } /* thin thick thin thick */
h1 { border-width: thin thick medium } /* thin thick medium thick */
```

8.5.2 Color del Borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', y 'border-color'

Las propiedades del color del borde especifican el color del borde de la caja.

'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color'

<i>Valor:</i>	<u><color></u> transparent <u>inherit</u>
<i>Inicial:</i>	El valor de la propiedad 'color'
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda</i>	no
<i>Porcentajes</i>	N/A
<i>Medio</i>	<u>visual</u>
<i>Valor Computado:</i>	Cuando lo toma de la propiedad 'color', el valor computado es 'color'; sino, el especificado

'border-color'

<i>Valor:</i>	[<u><color></u> transparent]{1,4} <u>inherit</u>
<i>Inicial:</i>	Ver propiedades individuales
<i>Se aplica a</i>	Todos los elementos
<i>Se hereda</i>	no
<i>Porcentaje</i>	N/A
<i>Medio</i>	<u>visual</u>
<i>Valor Computado</i>	Ver propiedades individuales

La propiedad 'border-color' determina el color de los cuatro bordes. Los valores tienen los siguientes significados:

<color>

Especifica un valor de color-

transparent

El borde es transparente (no obstante puede tener grosor).

La propiedad 'border-color' puede tener de uno a cuatro valores, y los valores son aplicados a los distintos lados como en 'border-width'.

Si el color del borde de un elemento no es especificado con una propiedad de borde, las aplicaciones del usuario deben tomar el valor de la propiedad 'color' del elemento como el [valor computado](#) para el color del borde.

En este ejemplo el borde será una línea negra sólida.

```
p {
  color: black;
  background: white;
  border: solid;
}
```


8.5.3 Estilo del borde: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style', y 'border-style'

Las propiedades del estilo del borde especifican el estilo de la línea del borde de una caja (sólida, doble, punteada, etc.). Las propiedades definidas en esta sección se refieren al tipo de valor de <border-style>, que puede estar constituido por uno de los siguientes:

none

ningún borde; el ancho del borde es cero.

hidden

Igual que 'none', excepto en términos de [resolución de conflictos de bordes](#) para los [elementos de tablas](#).

dotted

El borde es una serie de puntos.

dashed

El borde es una serie de pequeños segmentos de líneas.

solid

El borde es un único segmento de línea.

double

El borde son dos líneas sólidas. La suma de las dos líneas y el espacio entre ellas es igual al valor de 'border-width'.

groove

El borde luce como si estuviera tallado en el lienzo.

ridge

Lo opuesto a 'groove': el borde parece que estuviera sobresaliendo (en relieve) del lienzo.

inset

El borde hace que toda la caja luzca como si estuviera empotrada (hundida) en el lienzo.

outset

Lo contrario de 'inset': El borde ha que toda la caja parezca sobresalir (en relieve) del lienzo.

Todos los bordes son dibujados por encima del fondo de la caja. El color de los bordes dibujados con valores 'groove', 'ridge', 'inset', y 'outset' depende de las [propiedades de color del borde](#) de los elementos, pero las AUs pueden elegir sus propios algoritmos para calcular el color actual utilizado. Por ejemplo, si el 'border-color' tiene el valor 'silver', entonces una AU puede utilizar un degradado de colores de blanco a gris oscuro para indicar un borde inclinado.

'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style'

Valor: <border-style> | inherit

Inicial: none

Se aplica a: Todos los elementos

Se hereda: no

Porcentajes: N/A

Medio: visual

Valor Computado: Como el especificado

'border-style'

Valor:	<u><border-style></u> {1,4} <u>inherit</u>
Inicial:	Ver propiedades individuales
Se aplica a:	Todos los elementos
Se hereda:	no
Porcentaje:	N/A
Medio:	<u>visual</u>
Valor Computado:	Ver propiedades individuales

La propiedad 'border-style' determina el estilo de los cuatro bordes. Puede tener de uno a cuatro valores, y los valores son distribuidos para los distintos lados como en 'border-width' visto más arriba.

```
#xy34 { border-style: solid dotted }
```

En el ejemplo de arriba, los bordes horizontales serán 'solid' y los bordes verticales serán 'dotted'.

Como el valor inicial del estilo de borde es 'none', ningún borde será visible a menos que se establezca el estilo de borde.

8.5.4 Propiedades resumidas del borde: 'border-top', 'border-right', 'border-bottom', 'border-left', y 'border'

'border-top', 'border-right', 'border-bottom', 'border-left'

Valor:	[<u><border-width></u> <u><border-style></u> <u><'border-top-color'></u>] <u>inherit</u>
Inicial:	Ver propiedades individuales
Se aplica a:	Todos los elementos
Se hereda:	no
Porcentajes:	N/A
Medio:	<u>visual</u>
Valor Computado:	Ver propiedades individuales

Esta es una propiedad resumida para definir el ancho, el estilo y el color del borde superior, derecho, inferior e izquierdo de una caja.

```
h1 { border-bottom: thick solid red }
```

La regla de arriba determinará el ancho, el estilo y el color del borde **debajo** del elemento H1. Los valores omitidos son colocados en sus valores iniciales. Como la siguiente regla no especifica un color de borde, el borde tendrá el color especificado en la propiedad 'color':

```
H1 { border-bottom: thick solid }
```

'border'

Valor:	[<border-width> <border-style> <'border-top-color'>] inherit
Inicial:	Ver propiedades individuales
Se aplica a:	Todos los elementos
Se hereda:	no
Porcentajes:	N/A
Medio:	<u>visual</u>
Valor Computado:	Ver propiedades individuales

La propiedad 'border' es una propiedad resumida para colocar el mismo ancho, color y estilo a los cuatro bordes de una caja. A diferencia de las propiedades resumidas 'margin' y 'padding', la propiedad 'border' no puede definir diferentes valores para los cuatro bordes. Para eso, deben usarse una o más de las otras propiedades del borde.

Por ejemplo, la primera regla de abajo es equivalente a las cuatro reglas siguientes a esta:

```
p { border: solid red }
p {
  border-top: solid red;
  border-right: solid red;
  border-bottom: solid red;
  border-left: solid red
}
```

Como, hasta cierto punto, las propiedades tienen un funcionamiento que se superpone, el orden en que las reglas son especificadas es importante.

Considere este ejemplo:

```
blockquote {
  border: solid red;
  border-left: double;
  color: black;
}
```

En el ejemplo de arriba, el color del borde izquierdo es negro, mientras que los otros bordes son rojos. Esto se debe a la definición del ancho, estilo y color para 'border-left'. Como el valor del color no está dado en la propiedad 'border-left', éste será tomado de la propiedad 'color'. El hecho de que la propiedad 'color' esté puesta después que la propiedad 'border-left' no es relevante.

8.6 Modelo de caja para elementos en línea en contexto bidireccionales.

Para cada caja de línea, las Aus deben tomar las cajas en línea generadas para cada elemento y procesar los márgenes, bordes y rellenos en orden visual (no en orden lógico).

Cuando la propiedad 'direction' de los elementos es 'ltr', la caja generada más a la izquierda de la primera caja de línea en la cual el elemento aparece tiene el margen izquierdo, borde izquierdo y relleno izquierdo, y la caja generada más a la derecha en la última caja de línea en la cual aparece el elemento tiene el relleno derecho, el borde derecho y el margen derecho.

Cuando la propiedad 'direction' de los elementos es 'rtl', la primera caja generada más a la derecha en la caja de línea en la cual aparece el elemento tiene el relleno derecho, el borde derecho y el margen derecho, y la última caja generada más a la izquierda en la caja de línea en la que aparece el elemento tiene el margen izquierdo, el borde izquierdo y el relleno izquierdo.

9 Modelo de formato visual

Contenido

- [9.1 Introducción al modelo de formato visual](#)
 - [9.1.1 El acceso visual](#)
 - [9.1.2 bloques de contención](#)
- [9.2 Control de la generación de cajas](#)
 - [9.2.1 Elementos a nivel de bloque y cajas de bloque](#)
 - [Cajas de bloque anónimas](#)
 - [9.2.2 Elementos a nivel de línea y cajas en línea](#)
 - [Cajas en línea anónimas](#)
 - [9.2.3 Cajas run-in](#)
 - [9.2.4 La propiedad 'display'](#)
- [9.3 Esquemas de posicionamiento](#)
 - [9.3.1 Elección de un de posicionamiento: la propiedad 'position'](#)
 - [9.3.2 Desplazamiento de las cajas 'top', 'right', 'bottom', 'left'](#)
- [9.4 Flujo normal](#)
 - [9.4.1 Contexto del formato de bloque](#)
 - [9.4.2 Contexto del formato en línea](#)
 - [9.4.3 Posicionamiento relativo](#)
- [9.5 Flotantes](#)
 - [9.5.1 Posicionamiento del flotante: la propiedad 'float'](#)
 - [9.5.2 Control del flujo al costado del flotante: 'clear'](#)
- [9.6 Posicionamiento absoluto](#)
 - [9.6.1 Posicionamiento fijo](#)
- [9.7 Relaciones entre 'display', 'position', y 'float'](#)
- [9.8 Comparación entre flujo normal, flotantes, y posicionamiento absoluto](#)
 - [9.8.1 Flujo normal](#)
 - [9.8.2 Posicionamiento relativo](#)
 - [9.8.3 Caja flotante](#)
 - [9.8.4 Posicionamiento absoluto](#)
- [9.9 Presentación por capas](#)
 - [9.9.1 Especificar el nivel de apilado: la propiedad 'z-index'](#)
- [9.10 Dirección del texto: las propiedades 'direction' y 'unicode-bidi'](#)

9.1 Introducción al modelo de formato visual

Este capítulo y el siguiente describe el modelo de formato visual: cómo las aplicaciones del usuario procesan [la estructura del documento](#) para los [medios](#) visuales .

En el modelo de formato visual, cada elemento de la estructura del documento genera cero o más cajas según el [modelo de caja](#). La disposición de estas cajas se gobierna por:

- [dimensiones de la caja](#) y [tipo](#)
- [esquema de posicionamiento](#) (flujo normal, flotante, y posición absoluta).
- relaciones entre los elementos en [la estructura del documento](#).
- información externa (Ej., tamaño del acceso visual (viewport), dimensiones [intrínsecas](#) de las imágenes, etc.).

Las propiedades definidas en este capítulo y el siguiente se aplican a los [medios continuos](#) y a los [medios paginados](#). Sin embargo, los significados de las [propiedades del margen](#) varían cuando están aplicados a los medios paginados (véase el [modelo de paginación](#) para los detalles).

El modelo de formato visual no especifica todos los aspectos del formato (ej., no especifica un algoritmo para el espacio entre letras). Las [aplicaciones del usuario con conformidad](#) pueden tener diferentes comportamientos para esos formatos no cubiertos por esta especificación.

9.1.1 El acceso visual (viewport)

Las aplicaciones del usuario para los [medios continuos](#) generalmente ofrecen a los usuarios *un acceso visual* (la ventana u otra área de visualización en pantalla) a través del cual los usuarios consultan un documento. Las aplicaciones del usuario pueden cambiar la composición del documento cuando el acceso visual es redimensionado (véase el [bloque de contención inicial](#)).

Cuando el acceso visual (viewport) es más pequeño que el área del lienzo en la cual se procesa el documento, la aplicación del usuario debe ofrecer un mecanismo de desplazamiento. En la mayoría hay un acceso visual (viewport) por [lienzo](#), pero las aplicaciones del usuario pueden procesar más de un lienzo (es decir, que proporcione diversas vistas del mismo documento).

9.1.2 Bloques de contención

En CSS2.1, muchas posiciones y tamaños de cajas son calculados con respecto a los límites de una caja rectangular llamada *bloque de contención*. En general, las cajas generadas actúan como bloques de contención para las cajas descendientes; decimos que una caja "establece" el bloque de contención para sus descendientes. La frase "el bloque de contención de la caja" significa "el bloque de contención en el cuál la caja existe", no aquella que genera.

Cada caja tiene una posición dada con respecto a su bloque de contención, pero no está confinada por este bloque de contención; lo puede [desbordar](#).

Los [detalles](#) del modo en que las dimensiones de un bloque de contención son calculadas se describen en el [capítulo siguiente](#).

9.2 Control de la generación de cajas

Las siguientes secciones describen los tipos de cajas que pueden ser generadas en CSS2.1. Un tipo de caja afecta, en parte, su comportamiento en el modelo de formato visual. La propiedad '[display](#)', descrita abajo, especifica un tipo de caja.

9.2.1 Elementos a nivel de bloque y cajas de bloques

Los *elementos a nivel de bloque* son aquellos elementos del documento fuente que son tratados visualmente como bloques (ej., párrafos). Algunos valores de la propiedad '[display](#)' conforman un elemento a nivel de bloque: 'block', 'list-item', y 'run-in' (algunas veces; ver [cajas run-in](#)), y 'table'.

Los elementos a nivel de bloque generan (excepto para los elementos que muestran 'tablas', los cuales se describen en un capítulo más tarde) *una caja de bloque principal que solo contiene cajas de bloque o [cajas en línea](#)*. La caja de bloque principal establece el [bloque de contención](#) para las cajas descendiente y el contenido generado y es también la caja implicada en cada esquema de posicionamiento. Las cajas de bloque principales participan en un [contexto de formato de bloque](#).

Algunos elementos a nivel de bloque generan cajas adicionales fuera de la caja principal: elementos 'list-item'. Estas cajas adicionales se colocan con respecto a la caja principal.

Cajas de bloque anónimas

En un documento como este:

```
< DIV >  
  un texto  
  <P>Más texto  
</DIV >
```

(y asumiendo que DIV y P tienen ambos 'display: block'), DIV parece tener un contenido en línea y un contenido de bloque. Para facilitar el formato, asumimos que existe una caja de bloque anónima alrededor de "Un texto".

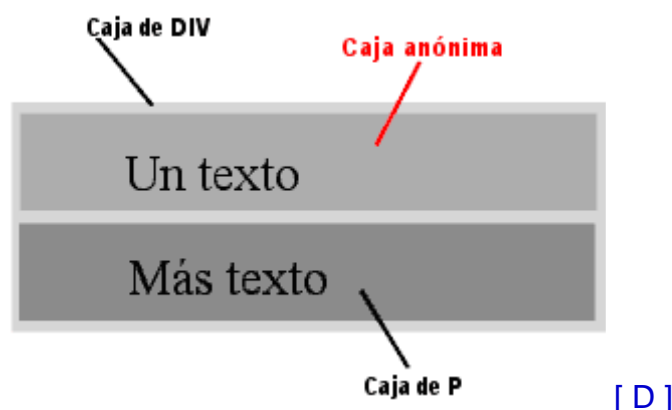


Diagrama mostrando las tres cajas, una de las cuales es anónima, para el ejemplo de arriba.

En otras palabras: si la caja de bloque (como las generadas por DIV, arriba) tiene otra caja de bloque en su interior (como la de P, arriba), entonces la forzamos a contener sólo cajas de bloque en su interior, envolviendo a las cajas en línea en una caja de bloque anónima..

Cuando una caja en línea contiene una caja de bloque, la caja en línea (y sus antepasados en línea dentro de la misma línea de la caja) está rota alrededor del bloque. Las cajas de líneas antes de la rotura y después de la rotura se encierran en cajas anónimas, y la caja de bloque se convierte en un hermano de esas cajas anónimas.

Este modelo se aplicaría en el siguiente ejemplo si las reglas siguientes:

```
body { display: inline }
p     { display: block }
```

fueran utilizadas con este documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HEAD>
<TITLE>texto anónimo interrumpido por un bloque</TITLE>
</HEAD>
<BODY>
Este es el texto anónimo antes de P.
<P>Este es el contenido de P</P>
Este es el texto anónimo después de P.
</BODY>
```

El elemento BODY contiene un segmento (C1) de texto anónimo seguido por un elemento a nivel de bloque seguido por otro segmento (C2) de texto anónimo. Las cajas resultantes serían una caja de bloque anónima para BODY, conteniendo una caja de bloque anónima alrededor de C1, la caja de bloque para P y otra caja de bloque anónima alrededor de C2.

Las propiedades de las cajas anónimas son heredadas de las cajas no-anónimas que las encierran (en el ejemplo: la de DIV). Las propiedades no heredadas tienen su valor inicial. Por ejemplo, la fuente de la caja anónima es heredada de DIV, pero los márgenes serán igual a 0.

Las propiedades fijadas en los elementos que causan bloques anónimos serán todavía generadas para aplicarlas a las cajas y contenidos de ese elemento. Por ejemplo, si un borde tiene que ser aplicado sobre el elemento BODY en el ejemplo anterior, el borde podría ser dibujado alrededor de C1 (abriéndose al final de la línea) y C2 (abriéndose al principio de la línea).

Algunas aplicaciones del usuario tienen implementado bordes sobre los bloques de contenidos en línea de diferente forma, por ejemplo, anidando tales bloques dentro de "líneas de bloques anónimos" y así dibujando bordes en línea alrededor de esas cajas. CSS1 y CSS2 no definieron este comportamiento, las aplicaciones del usuario de CSS1-solo y CSS2-solo pueden implementar este modelo alternativo y todavía reclaman conformidad para esta parte del CSS2.1. Esto no se aplica a AUs desarrolladas después de que esta especificación fuera lanzada.

9.2.2 Elementos a nivel de línea y cajas en línea

Los *elementos a nivel de línea* son aquellos elementos del documento fuente que no forman nuevos bloques de contenido; el contenido es distribuido en líneas (ej., fragmentos de texto enfatizado dentro de un párrafo, imágenes en línea, etc.). Algunos valores de la propiedad '[display](#)' conforman un elemento en línea: 'inline', 'inline-table' y 'run-in' (algunas veces; ver [cajas run-in](#)). Los elementos a nivel de línea generan cajas en línea.

Cajas anónimas en línea

En un documento HTML como éste:

```
<p>Algún <em>texto</em> enfatizado</p>
```

`<p>` genera una caja de bloque, con algunas cajas en línea dentro de ella. La caja "texto" es una caja en línea generada por un elemento en línea (``), pero las otras cajas ("Algún" y "enfatizado") son cajas en línea generadas por un elemento a nivel de bloque (`<p>`). Los últimos son denominados cajas de línea anónimas porque no tienen un elemento a nivel de línea asociado.

Tales cajas anónimas en línea heredan las propiedades hereditarias de su caja de bloque padre. Las propiedades no hereditarias tienen su valor inicial. En el ejemplo, el color de las cajas en línea anónimas es heredado de P, pero el fondo es transparente.

Los espacios en blanco del contenido que posteriormente podrán ser cerrados de acuerdo a la propiedad `'white-space'` (espacio en blanco) no genera ninguna caja anónima en línea.

Si queda claro en el contexto a qué tipo de caja anónima se refiere, las cajas en línea anónimas y las cajas de bloque anónimas son denominadas simplemente cajas anónimas en esta especificación.

Hay más tipos de cajas anónimas que surgen en el formato de las [tablas](#).

9.2.3 Cajas run-in

Una caja *run-in* se comporta como sigue:

1. Si la caja run-in contiene una caja de [bloque](#), la caja run-in se convierte en una caja de bloque.
2. Si una caja de [bloque](#) hermana (que no sea flotante y no esté [posicionada absolutamente](#)) sigue a la caja run-in, la caja run-in se convierte en la primera caja en línea de la caja de bloque. Una run-in no puede funcionar dentro a un bloque que comience ya con una run-in o ella misma sea una run-in.
3. Si no, la caja run-in se convierte en una caja de bloque.

Una caja 'run-in' es útil para los encabezados run-in, como en este ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Un ejemplo de caja run-in</TITLE>
    <STYLE type="text/css">
      H3 { display: run-in }
    </STYLE>
  </HEAD>
  <BODY>
    <H3>Un encabezado run-in.</H3>
    <P>Y un párrafo de texto que lo sigue.
  </BODY>
</HTML>
```

Este ejemplo podría ser procesado como:

Un encabezado run-in. Y un párrafo de texto que lo sigue

Las propiedades del elemento run-in son heredados de su padre en la estructura del documento, no de la caja de bloque de la cual visualmente se vuelve parte.

Consulte por favor la sección sobre [contenido generado](#) para la información acerca de cómo las cajas run-in interactúan con el contenido generado.

9.2.4 La propiedad '[display](#)'

'display'

<i>Valor:</i>	inline block list-item run-in inline-block table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption none inherit
<i>Inicial:</i>	inline
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	all
<i>Valor Computado</i>	Ver texto

Los valores de esta propiedad tienen los siguientes significados:

block

Este valor provoca que un elemento genere una caja de bloque principal.

Inline-block

Este valor provoca que un elemento genere una caja de bloque, la cual fluye como una simple caja en línea, similar a un elemento reemplazado. El interior de un bloque en línea está formado por una caja de bloque y este elemento se ajusta a un formato de elemento en línea reemplazado.

inlínea

Este valor provoca que un elemento genere una o mas cajas en línea.

List-item

Este valor provoca que un elemento (ej., LI en HTML) genere una caja de bloque principal y una caja en línea list-item. Para información sobre listas y ejemplos del formato de la lista, consulte por favor la sección sobre [listas](#).

none

Este valor provoca que un elemento no genere cajas en la [estructura de formato](#) (es decir, el elemento no tiene ningún efecto en la disposición). Los elementos descendientes no generan ninguna caja tampoco; este comportamiento **no puede** ser eliminado fijando la propiedad '[display](#)' en los descendientes.

Por favor observe que un display con valor 'none' no crea una caja invisible, esto no crea ninguna caja en absoluto. CSS incluye mecanismos que permite a un elemento generar cajas en la estructura de formato que afectan a la composición pero ellas mismas no son visibles. Por favor consulte la sección sobre [visibilidad](#) para los detalles.

run-in

Este valor crea cajas de bloque o cajas en línea, dependiendo del contexto. Las propiedades se aplican a cajas run-in en base a su estatus final (a nivel de línea o a nivel de bloque).

table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, and table-caption

Estos valores provocan que un elemento se comporte como un elemento tabla (sujeto a restricciones descritas en el capítulo sobre [tablas](#)).

El valor computado es el mismo que el valor especificado, excepto para los elementos posicionados y flotantes (Vea [Relaciones entre 'display', 'position', y 'float'](#)) y para el elemento raíz. Para el elemento raíz, el valor computado es cambiado como se describe en la sección sobre [relaciones entre 'display', 'position', y 'float'](#).

Observe que aunque el [valor inicial](#) de 'display' es 'inline', las reglas de la [hoja de estilo predeterminada](#) de la aplicación del usuario puede [sustituir](#) este valor. Ver la [hoja de estilo de ejemplo](#) para HTML 4.0 en el apéndice.

Aquí hay algunos ejemplos de la propiedad ['display'](#):

```
p { display: block }
em { display: inline }
li { display: list-item }
img { display: none } /* no muestra imagenes */
```

9.3 Esquemas de posicionamiento

En CSS2.1, una caja puede ser presentada de acuerdo a tres esquemas de posicionamiento:

1. [Flujo normal](#) En CSS2.1, el flujo normal incluye [formato de bloque](#) de [cajas de bloque](#), [formatos en línea](#) de [cajas en línea](#), [posicionamiento relativo](#) de cajas de bloque o en línea, y posicionamiento de cajas [run-in](#).
2. [Flotantes](#). En el modelo flotante, una caja se presenta primero de acuerdo al flujo normal, luego se saca del flujo normal y se mueve a la izquierda o derecha tanto como sea posible. El contenido puede fluir a lo largo del costado del flotante.
3. [Posicionamiento absoluto](#). En el modelo de posicionamiento absoluto, una caja es quitada completamente del flujo normal (no tiene ningún impacto sobre los hermanos siguientes) y se le asigna una posición con respecto al bloque de contención.

Nota. *Los esquemas de posicionamiento de CSS2.1 ayudan a que los autores hagan sus documentos más accesibles permitiéndoles evitar los artilugios (ej., imágenes invisibles) usados para los efectos de composición.*

9.3.1 Elección de un esquema de posicionamiento: la propiedad **'position'**

Las propiedades **'position'** y **'float'** determinan que algoritmo de posicionamiento CSS2.1 es utilizado para calcular la posición de la caja.

'position'

<i>Valor:</i>	static relative absolute fixed <u>inherit</u>
<i>Inicial:</i>	static
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no
<i>Porcentaje:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

Los valores de esta propiedad tienen los siguientes significados:

static

La caja es una caja normal, presentada de acuerdo al [flujo normal](#). Las propiedades **'top'**, **'right'**, **'bottom'**, y **'left'** no se aplican.

relative

La posición de la caja es calculada de acuerdo al [flujo normal](#) (ésta es llamada la posición en el flujo normal). Luego la caja es desplazada de modo [relativo](#) a su posición normal. Cuando una caja B es posicionada relativamente, la posición de la siguiente caja es calculada como si B no se hubiera desplazado. El efecto de **'position:relative'** sobre los elementos **table-row-group** (grupo de fila de la tabla), **table-header-group** (grupo de cabecera de la tabla), **table-footer-group** (grupo de apilamiento de la tabla), **table-row** (fila de la tabla), **table-column-group** (grupo de columna de la tabla), **table-column** (columna de la tabla), **table-cell** (celda de la tabla), y **table-caption** (titulo de la tabla) son indefinidos.

absolute

La posición de la caja (y posiblemente el tamaño) se especifican con las propiedades **'top'**, **'right'**, **'bottom'**, y **'left'**. Estas propiedades especifican los desplazamientos con respecto al [bloque de contención](#) de la caja. Las cajas posicionadas Absolutamente son quitadas del flujo normal. Esto significa que éstas no tienen ningún impacto sobre la composición de los hermanos siguientes. También, aunque las cajas [posicionadas absolutamente](#) tienen márgenes, estos no se [cierran](#) con ningún otro margen.

fixed

La posición de la caja es calculada de acuerdo al modelo **'absolute'**, pero además, la caja se [fija](#) con respecto a alguna referencia. Al igual que en el modelo **'absolute'**, los márgenes de la caja no se cierran con ningún otro margen. En el caso de tipos de medios manuales, de proyección, monitores, tty, y tv, la caja es fijada con respecto al [acceso visual](#) y no se mueve cuando se realiza un desplazamiento (mover la barra de desplazamiento del navegador). En el caso de tipos de medios de impresión, la caja es procesada sobre cada página, y es fijada con respecto a la página, incluso si la página es visualizada a través de un [acceso visual](#) (en el caso de la vista preliminar para imprimir, por ejemplo). Para otros tipos de medios, la presentación es indefinida. Los autores pueden desear especificar **'fixed'** de manera dependiente de los medios. Por ejemplo, un autor puede desear que una caja permanezca en la parte superior del [acceso visual](#) en la pantalla, pero no al comienzo de la página de impresión. Las dos especificaciones pueden ser separadas utilizando una [regla @media](#), como en:

```

@media screen {
  h1#first { position: fixed }
}
@media print {
  h1#first { position: static }
}

```

Las aplicaciones del usuario pueden tratar la posición como 'static' sobre el elemento raíz.

9.3.2 Desplazamiento de las cajas: **'top'**, **'right'**, **'bottom'**, **'left'**

Se dice que un elemento está posicionado si su propiedad 'position' tiene un valor distinto a 'static'. Los elemento posicionados generan cajas posicionadas, situadas de acuerdo a cuatro propiedad:

'top'

Valor: <medida> | <porcentaje> | auto | inherit
Inicial: auto
Se aplica a: Elementos posicionados
Se hereda: no
Porcentajes: Referidos a la altura del bloque de contención
Medio: visual
Valor computado Para 'position:relative', ver la sección [Posición Relativa](#). Para 'position:static', es 'auto'. Si no: si se especifica como una medida, la correspondiente medida absoluta, si se especifica como un porcentaje, el valor especificado; si no 'auto'.

Esta propiedad especifica cuán lejos se desplaza el limite (borde) del margen superior de una caja [posicionada absolutamente](#) por debajo del limite (borde) superior del [bloque de contención](#).de la caja. Para cajas posicionadas relativamente, el desplazamiento es con respecto al limite del margen superior de ella misma (es decir, se le da a la caja una posición dentro del flujo normal, luego se desplaza de esa posición de acuerdo a estas). *Nota: Para el posicionamiento absoluto cuyo bloque de contención está basado sobre un elemento a nivel de bloque, esta propiedad es un desplazamiento del limite del relleno de ese elemento.*

'right'

Valor: <medida> | <porcentaje> | auto | inherit
Inicial: auto
Se aplica a: Elementos posicionados
Se hereda: no
Porcentajes: Referidos a la altura del bloque de contención
Medio: visual
Valor computado Para 'position:relative', ver la sección [Posición Relativa](#). Para 'position:static', es 'auto'. Si no: si se especifica como una medida, la correspondiente medida absoluta, si se especifica como un porcentaje, el valor especificado; si no 'auto'.

Igual que 'top', pero especifica cuán lejos se desplaza el límite (borde) del margen derecho de una caja hacia la izquierda del límite (borde) derecho del [bloque de contención](#).de la caja. Para cajas posicionadas relativamente, el desplazamiento es con respecto al límite derecho de ella misma. **Nota:** Para el posicionamiento absoluto cuyo bloque de contención está basado sobre un elemento a nivel de bloque, esta propiedad es un desplazamiento del límite (borde) del relleno de ese elemento

'bottom'

Valor: [<medida>](#) | [<porcentaje>](#) | auto | [inherit](#)
Inicial: auto
Se aplica a: Elementos posicionados
Se hereda: no
Porcentajes: Referidos a la altura del bloque de contención
Medio: [visual](#)
Valor computado Para 'position:relative', ver la sección [Posición Relativa](#). Para 'position:static', es 'auto'. Si no: si se especifica como una medida, la correspondiente medida absoluta, si se especifica como un porcentaje, el valor especificado; si no 'auto'.

Igual que 'top', pero especifica cuán lejos se desplaza el límite (borde) del margen inferior derecho de una caja hacia arriba del límite (borde) inferior del [bloque de contención](#).de la caja. Para cajas posicionadas relativamente, el desplazamiento es con respecto al límite inferior de ella misma. **Nota:** Para el posicionamiento absoluto cuyo bloque de contención está basado sobre un elemento a nivel de bloque, esta propiedad es un desplazamiento del límite del relleno de ese elemento

'left'

Valor: [<medida>](#) | [<porcentaje>](#) | auto | [inherit](#)
Inicial: auto
Se aplica a: Elementos posicionados
Se hereda: no
Porcentajes: Referidos a la altura del bloque de contención
Medio: [visual](#)
Valor computado Para 'position:relative', ver la sección [Posición Relativa](#). Para 'position:static', es 'auto'. Si no: si se especifica como una medida, la correspondiente medida absoluta, si se especifica como un porcentaje, el valor especificado; si no 'auto'.

Igual que 'top', pero especifica cuán lejos se desplaza el límite (borde) del margen izquierdo de una caja hacia la derecha del límite (borde) izquierdo del [bloque de contención](#).de la caja. Para cajas posicionadas relativamente, el desplazamiento es con respecto al límite izquierdo de ella misma. **Nota:** Para el posicionamiento absoluto cuyo bloque de contención está basado sobre un elemento a nivel de bloque, esta propiedad es un desplazamiento del límite del relleno de ese elemento

Los valores para las cuatro propiedades tienen los siguientes significados:

<medida>

El desplazamiento es una distancia fija desde el límite (borde) de referencia. Los valores negativos son permitidos.

<porcentaje>

El desplazamiento es un porcentaje del ancho (para 'left' o 'right') o del alto (para 'top' and 'bottom') del bloque de contención. Los valores negativos son permitidos.

auto

Para los elementos no reemplazados (non-replaced), el efecto de este valor depende de que propiedades relacionadas tienen también el valor 'auto'. Ver la sección sobre [ancho](#) y [altura](#) de los elementos no reemplazados [posicionados absolutamente](#) para más detalles. Para elementos sustituidos, el efecto de este valor depende solamente de la dimensión intrínseca del contenido sustituido. Ver la sección sobre [ancho](#) y [altura](#) de los elementos sustituidos [posicionados absolutamente](#) para más detalles.

9.4 Flujo normal

Las cajas dentro del flujo normal pertenecen al contexto del formato, que puede ser de bloque o en línea, pero no ambos simultáneamente. Las cajas de [bloque](#) participan en un contexto de [formato de bloque](#). Las [cajas en línea](#) participan en un contexto de [formato en línea](#).

9.4.1 Contexto de formato de bloque

Flotantes, elementos posicionados absolutamente, bloques en línea (inline-blocks), celdas de la tabla (table-cells), y elementos con 'overflow' (desbordamiento) diferente a 'visible' establecen nuevos contextos de formato de bloque.

En un contexto de formato de bloque, las cajas se colocan una después de otra, verticalmente, comenzando desde lo alto de un bloque de contención. La distancia vertical entre dos cajas hermanas es determinada por las propiedades 'margin'. Los márgenes verticales entre cajas de bloque adyacentes en un contexto de formato de bloque se [cierran](#).

En un contexto de formato de bloque, cada límite (borde) izquierdo externo de la caja toca el límite (borde) izquierdo del bloque de contención (para la composición de derecha a izquierda, los bordes derechos se tocan). Esto es así aún en presencia de flotantes (aunque el área del contenido de una caja puede encogerse debido a las flotantes), a no ser que la caja establezca un nuevo contexto de formato de bloque (en tal caso la caja puede hacerse más estrecha debido a las flotantes).

Para información acerca de los saltos de página en los medios paginados, por favor consulte la sección sobre [saltos de página permitidos](#).

9.4.2 Contexto de formato en línea

En un contexto de formato en línea, las cajas son colocadas horizontalmente, una después de otra, comenzando desde lo alto de un bloque de contención. Los márgenes horizontales, bordes y rellenos son respetados entre estas cajas. Las cajas pueden ser alineadas verticalmente de diferentes maneras: pueden alinearse por su parte inferior o superior, o pueden alinearse por las líneas de base del texto en su interior. El área rectangular que contiene las cajas que forman una línea son llamadas *cajas en línea*.

El ancho de una caja en línea es determinado por un [bloque de contención](#) y la presencia de flotantes. La altura de una caja en línea es determinada por las reglas establecidas en la sección [cálculo de la altura de la línea](#).

Una caja en línea es siempre lo suficientemente alta para todas las cajas que contiene. Sin embargo, puede ser mayor que la caja más alta que contiene (si, por ejemplo las cajas están alineadas de manera que sus líneas de base queden en línea). Cuando la altura de una caja B es menor que la altura de la caja en línea que la contiene, la alineación vertical de B dentro de la caja en línea es determinada por la propiedad 'vertical-align' (alineación vertical). Cuando varias cajas en línea no pueden ajustarse horizontalmente dentro de una sola caja en línea, éstas son distribuidas entre dos o más cajas en línea apiladas verticalmente. De este modo, un párrafo es una pila vertical de cajas en línea. Las cajas en línea son apiladas sin separación vertical y nunca se superponen.

En general, el límite (borde) izquierdo de una caja en línea toca el límite (borde) izquierdo del bloque de contenido y el límite (borde) derecho toca el límite (borde) derecho de su bloque de contención. Sin embargo, las cajas flotantes pueden interponerse entre el límite (borde) del bloque de contención y el límite (borde) de la caja en línea. Así, aunque las cajas en línea en el mismo contexto de formato en línea generalmente tienen el mismo ancho (el del bloque de contención), éstas pueden variar en su ancho si el espacio horizontal es reducido por las [flotantes](#). Las cajas en línea en el mismo contexto de formato en línea pueden variar su altura (ej., Una línea podría contener una imagen muy alta mientras que las otras contienen solo texto).

Cuando el ancho total de las cajas en línea en una línea es menor que el ancho de la caja en línea que las contiene, su distribución horizontal dentro de la caja de línea es determinada por la propiedad '[text-align](#)' (alineación del texto). Si esa propiedad tiene el valor 'justify', la aplicación del usuario puede estirar también las cajas de línea.

Cuando una caja en línea excede el ancho de una línea de caja, ésta es dividida en varias cajas y estas cajas son distribuidas en varias cajas en línea. Si una caja en línea no puede ser dividida (por ejemplo, si la caja en línea contiene un carácter simple o una palabra específica de un lenguaje que rompe las reglas rechazando la partición dentro de la caja en línea, o si la caja en línea es afectada por valores de espacios en blanco nowrap o pre), entonces la caja en línea desborda la línea de la caja.

Cuando una caja en línea es dividida, los márgenes, bordes y rellenos no tienen ningún efecto visual donde se produce la división (o en cualquier división, cuando hay varias).

Las cajas en línea también pueden dividirse en varias cajas dentro de la misma línea de la caja debido al [procesamiento del texto bidireccional](#).

Aquí hay un ejemplo de construcción de una caja en línea. El siguiente párrafo (creado por el elemento HTML a nivel de bloque P) contiene texto anónimo mezclado con los elementos EM y STRONG:

```
<P>Algunas <EM>palabras con énfasis</EM> aparecen
<STRONG>en esta</STRONG> oración, querida.</P>
```

El elemento P genera una caja de bloque que contiene cinco cajas en línea, tres de las cuáles son anónimas:

- Anónimo: "Algunas"
- EM: "palabras con énfasis"
- Anónimo: "aparecen"
- STRONG: "en esta"
- Anónimo: "oración, querida."

Para dar formato al párrafo, la aplicación del usuario fluye las cinco cajas dentro de cajas en línea. En este ejemplo, la caja generada por el elemento P establece el bloque de contención para las cajas en línea. Si el bloque de contención es suficientemente ancho, todas las cajas en línea encajarán en una sola caja en línea:

```
Algunas palabras con énfasis aparecen en esta oración, querida.
```

Si no, las cajas en línea serán divididas y distribuidas en varias cajas en línea. El párrafo anterior podría dividirse como sigue:

```
Algunas palabras con énfasis aparecen
en esta oración, querida.
```

o así:

```
Algunas palabras
con énfasis aparecen
en esta oración,
querida.
```

En el ejemplo anterior, la caja EM se dividió en dos cajas EM (llamémosla "split1" y "split2"). Los márgenes, bordes y relleno, o las decoraciones del texto no tiene efecto visible después de split1 o antes de split2.

Considere el ejemplo siguiente:

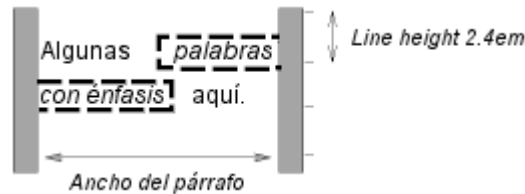
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"><HTML>
<HEAD>
  <TITLE>Ejemplo de flujo en línea sobre varias líneas</TITLE>
  <STYLE type="text/css">
    EM {
      padding: 2px;
      margin: 1em;
      border-width: medium;
      border-style: dashed;
      line-height: 2.4em;
    }
  </STYLE>
</HEAD>
<EM>
  Algunas palabras
  con énfasis aparecen
  en esta oración,
  querida.
</EM>
```

```

    }
  </STYLE>
</HEAD>
<BODY>
  <P>Algunas <EM>palabras con énfasis</EM> aquí.</P>
</BODY>
</HTML>

```

Dependiendo del ancho de P, las cajas pueden distribuirse como sigue:



- El margen es insertado antes de “palabras” y después de “con énfasis”.
- El relleno se inserta antes, sobre, y debajo de “palabras” y después, sobre y debajo de “con énfasis”. Un borde con líneas discontinuas es procesado en los tres lados en cada caso.

9.4.3 Posicionamiento relativo

Una vez que la caja es situada de acuerdo al [flujo normal](#) o flotante, puede moverse con relación a esta posición. A esto se le llama posicionamiento relativo. Desplazando una caja (B1) de esta manera no conlleva ningún efecto sobre la caja (B2) que le sigue: a B2 se le da una posición como si B1 no hubiera sido desplazada y B2 no es reposicionada después de que el desplazamiento de B1 ha sido aplicado. Esto implica que el posicionamiento relativo puede ocasionar que las cajas se superpongan. Sin embargo, si el posicionamiento relativo causa un ‘overflow:auto’ (desbordamiento:auto) en la caja que tiene desbordamiento, la AU puede permitir al usuario acceder a este contenido, el cual, con la creación de barras de desplazamiento (scrollbars), puede afectar la disposición

Las cajas posicionadas relativamente mantienen su tamaño de flujo normal, incluyendo los saltos de línea y el espacio originalmente reservado para ellos. La sección sobre [bloques de contención](#) explica cuando una caja posicionada relativamente establece un nuevo bloque de contención.

Para un elemento posicionado relativamente, 'left' (izquierda) y 'right' (derecha) desplazan la(s) caja(s) horizontalmente, sin cambiar su tamaño. 'left' (izquierda) desplazan las cajas a la derecha, y entonces 'right' (derecha) las desplaza a la izquierda. Ya que las cajas no son divididas o estiradas como resultado de 'left' o 'right', los valores computados son siempre: $left = -right$.

Si ambos 'left' (izquierda) y 'right' (derecha) son 'auto' (su valor inicial), los valores computados son '0' (es decir, las cajas se quedan en su posición original).

Si 'left' (izquierda) es 'auto', su valor computado es disminuido del valor de 'right' (derecho) (es decir, la caja se desplaza a la izquierda por el valor de 'right' (derecho)).

Si 'right' (derecha) es especificado como 'auto', su valor computado es disminuido del valor de 'left' (izquierda).

Si ni 'left' (izquierda) ni 'right' (derecha) son 'auto', la posición es sobre obligada, y uno de ellos tiene que prevalecer. Si la propiedad 'direction' es 'ltr', el valor de 'left' gana y 'right' se convierte en '-left'. Si 'direction' es 'rtl', 'right' gana y 'left' es ignorado.

Las siguientes tres reglas son equivalentes:

```
div.a8 { position: relative; direction: ltr; left: -1em; right: auto }
div.a8 { position: relative; direction: ltr; left: auto; right: 1em }
div.a8 { position: relative; direction: ltr; left: -1em; right: 5em }
```

Las propiedades 'top' (superior) y 'bottom' (inferior) de un elemento posicionado relativamente lo desplazan arriba y abajo sin modifica su tamaño. 'top' desplaza las cajas hacia abajo y 'bottom' hacia arriba. Ya que las cajas no son divididas o estiradas como resultado de 'top' o 'bottom', los valores computados son siempre: top = -bottom. Si ambos son 'auto', sus valores computados para ambos es '0'. Si uno de ellos es 'auto', el otro se convierte en negativo. Si ninguno es 'auto', 'bottom' es ignorado (es decir, el valor computado de 'bottom' será disminuido por el valor de 'top').

Nota. El movimiento dinámico de cajas posicionadas relativamente pueden producir efectos de animación en entornos de scripting (Ver también la propiedad 'visibility'). El posicionamiento relativo también puede utilizarse como una forma de generar súper scripting y subscripting, la altura de la línea no es ajustada automáticamente para tomar en consideración el posicionamiento Ver la descripción de [cálculo de la altura de la línea](#) para más información.

Ejemplos de posicionamientos relativos se proporcionan en la sección [comparación entre flujo normal, flotantes y posicionamiento absoluto](#).

9.5 Flotantes

Una flotante es una caja que es desplazada a la izquierda o la derecha sobre la línea actual. La característica más interesante de una flotante (o “flotador” o caja “flotante”) es que su contenido puede fluir a lo largo de su costado (o prohibir hacerlo mediante la propiedad 'clear'). El contenido fluye hacia abajo del costado derecho de un caja flotante a la izquierda y por el lado izquierdo de una caja flotante a la derecha. Lo siguiente es una introducción al posicionamiento flotante y al flujo del contenido; las [reglas](#) exactas que gobiernan el comportamiento flotante se dan en la descripción de la propiedad 'float'.

Una caja es desplazada a la izquierda o derecha antes de que su borde externo toque el bloque de contenido o el borde externo de otra flotante. Si hay una caja en línea la parte superior de la caja flotante es alineada con la parte superior de la actual línea de la caja.

Si no hay suficiente espacio horizontal para la flotante, esta es desplazada hacia abajo hasta que quepa o no haya presente más flotantes.

Puesto que una flotante no está en el flujo, las cajas de bloque no posicionadas creadas antes y después de la caja flotante fluyen verticalmente como si la flotante no existiera. Sin embargo, las cajas en línea creadas al lado de la flotante son acortadas para darle espacio a la caja flotante. Si la caja en línea acortada es muy pequeña para contener algún otro contenido, entonces esta es desplazada hacia abajo hasta que haya espacio o no haya presente mas flotantes. En otras palabras, si las cajas en línea son colocadas sobre la línea antes de que la flotante izquierda encuentre sitio en el espacio restante de la línea de la caja, la flotante izquierda es situada sobre esa línea, alineada con el margen superior de la caja en línea, y entonces las cajas en línea sobre esa línea son movidas a la parte derecha de la flotante (la derecha empieza al otro lado de la flotante izquierda) y viceversa para rtl y flotantes derechas.

El margen de la caja de una tabla o de un elemento en el flujo normal que establece un nuevo [contexto de formato de bloque](#) (así como un elemento con 'overflow' como 'visible') no debe superponer ninguna flotante en el mismo contexto de formato como al elemento. Si es necesario, las aplicaciones puede limpiar dicho elemento colocándolo debajo de alguna flotante precedente, pero puede colocarlos como adyacentes a las flotantes si hay espacio suficiente.

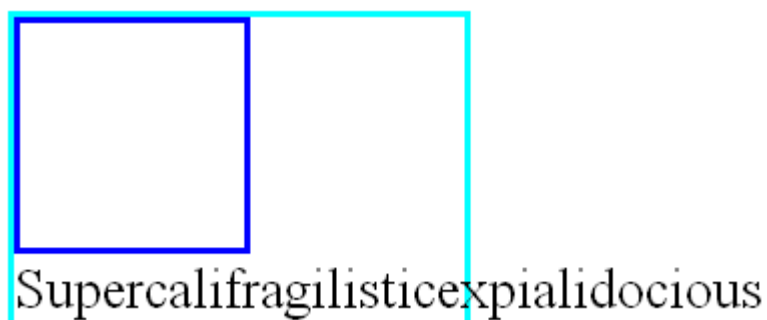
En el siguiente fragmento de documento, el bloque de contención es muy estrecho para contener el contenido de la siguiente flotante, así que el contenido es desplazado debajo de la flotante donde es alineada en la línea de la caja de acuerdo a la propiedad text-align

```
p { width: 10em; border: solid aqua; }
span { float: left; width: 5em; height: 5em; border: solid blue; }

...

<p>
  <span> </span>
  Supercalifragilisticexpialidocious
</p>
```

Este fragmento podría parecerse a esto:



Varias flotantes pueden ser adyacentes, y este modelo también se aplica a flotantes adyacentes en la misma línea.

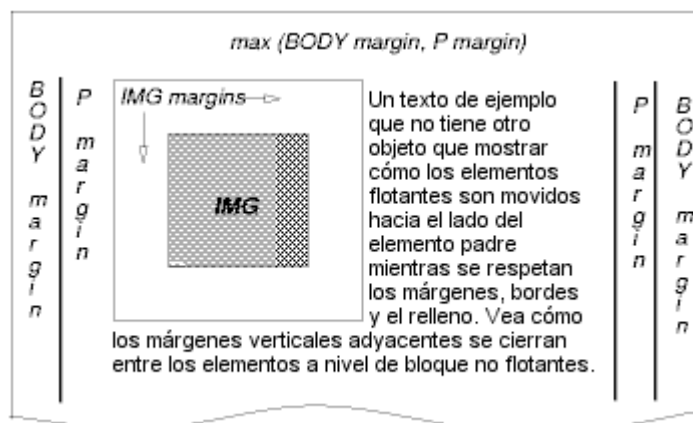
La siguiente regla hace flotar todas las cajas IMG con `class="icon"` a la izquierda (y pone el margen izquierdo a '0'):

```
img.icon {
  float: left;
  margin-left: 0;
}
```

Considere el siguiente código fuente HTML y hoja de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Float example</TITLE>
    <STYLE type="text/css">
      IMG { float: left }
      BODY, P, IMG { margin: 2em }
    </STYLE>
  </HEAD>
  <BODY>
    <P><IMG src=img.png alt="Esta imagen ilustrará un flotante">
      Un texto de ejemplo que no tiene otro...
    </P>
  </BODY>
</HTML>
```

La caja IMG flota hacia la izquierda. El contenido que le sigue es procesado a la derecha del flotante, comenzando en la misma línea que el flotante. Las cajas de línea a la derecha del flotante son acortadas debido a la presencia del flotante, pero retoman su ancho "normal" (el del bloque de contención establecido por el elemento P) después del flotante. Este documento podría procesarse como:



[D]

El formato habría sido exactamente el mismo si el documento hubiera sido:

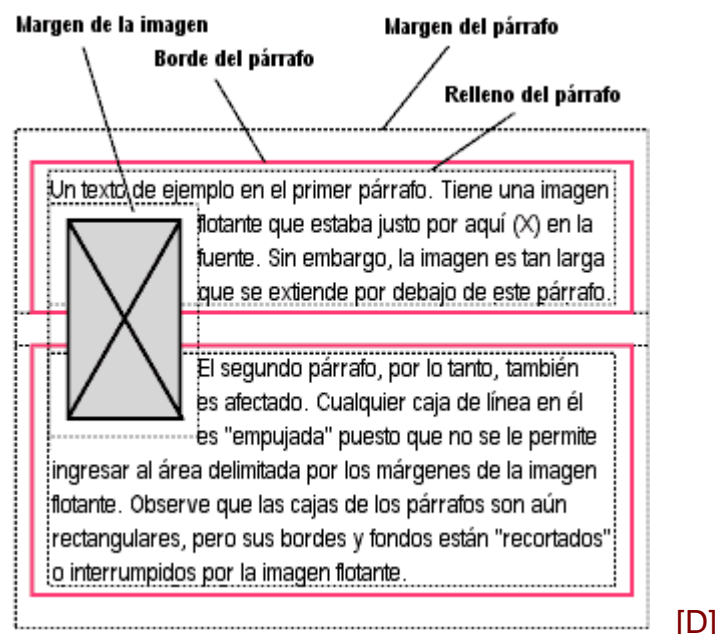
```
<BODY>
  <P>Some sample text
  <IMG src=img.png alt="Esta imagen ilustrará un flotante">
    Que no tiene otro...
</BODY>
```

debido a que el contenido a la izquierda del flotante es desplazado por el flotante y vuelto a fluir por su lado derecho.

Como se comento en la [sección 8.3.1](#), los márgenes de las cajas flotantes nunca se [cierran](#) con los márgenes de las cajas adyacentes. De este modo, en el ejemplo anterior, los márgenes verticales no se [cierran](#) entre la caja P y la caja flotante IMG.

Los contenidos de las flotantes son apilados como si las flotantes generaran nuevos contextos apilados, excepto que algunos elementos que crean nuevos contextos apilados toman parte de los contextos apilados de las flotantes padres. Una flotante puede superponerse a otras cajas en el flujo normal (ej., Cuando una caja de flujo normal al lado de una flotante tiene márgenes negativos). Cuando esto ocurre, las flotantes son procesadas delante en el flujo de bloques no posicionados, pero detrás en el flujo de líneas.

Aquí hay otra ilustración que muestra lo que sucede cuando una flotante se superpone con los bordes de elementos en el flujo normal.



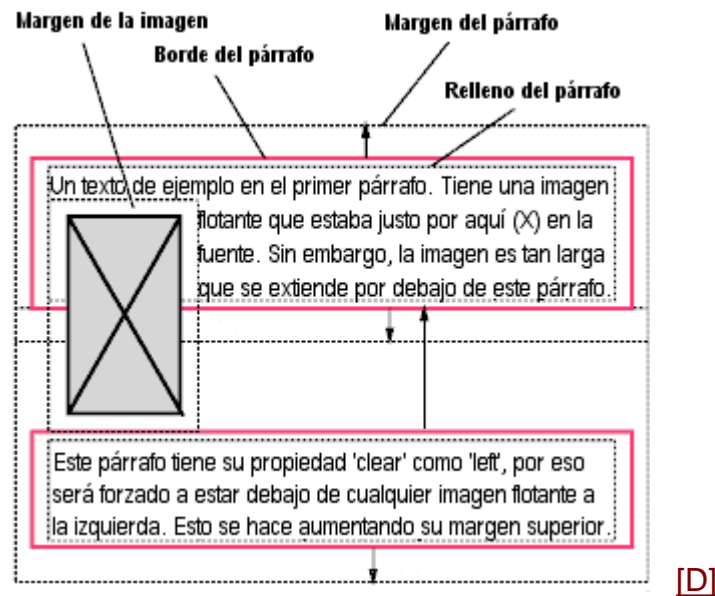
Una imagen flotante oculta los bordes de las cajas de bloque con las que se superpone.

El ejemplo siguiente ilustra el uso de la propiedad ['clear'](#) para impedir que el contenido fluya al costado de una flotante

Asumiendo una regla como esta:

```
p { clear: left }
```

el formato podría parecerse a esto:



*Ambos párrafos tienen especificado 'clear: left', lo que provoca que el segundo párrafo sea "empujado" a una posición debajo del flotante --su margen superior se expande para conseguir esto (ver la propiedad **'clear'**).*

9.5.1 Posicionamiento del flotante: la propiedad **'float'**

'float'

Valor: left | right | none | inherit
Inicial: none
Se aplica a: Todos, pero ver [9.7](#)
Se hereda: no
Porcentajes: N/A
Medios: visual
Valor computado: Como el especificado

Esta propiedad especifica si una caja debe flotar a la izquierda, derecha o no debe flotar en absoluto. Puede especificarse para los elementos que generan cajas que no están [posicionadas absolutamente](#). Los valores de esta propiedad tienen los siguientes significados:

left

El elemento genera una caja de [bloque](#) que flota a la izquierda. El contenido fluye sobre el lado derecho de la caja, comenzando en la parte superior (sujeto a la propiedad 'clear').

right

Igual a 'left', excepto que la caja flota a la derecha, y el contenido fluye sobre el lado izquierdo de la caja, comenzando en la parte superior.

none

La caja no flota.

Las aplicaciones del usuario pueden tratar float como 'none' sobre el elemento raíz.

Aquí están las reglas precisas que gobiernan el comportamiento de los flotantes:

1. El [borde exterior](#) izquierdo de una caja flotante a la izquierda no puede estar a la izquierda del borde izquierdo de su [bloque de contención](#). Una regla análoga se establece para los elementos flotantes a la derecha.
2. Si la caja actual es flotante a la izquierda y hay otra caja flotante a la izquierda generada por elementos que aparecen antes en el documento fuente, entonces para cada una de tales cajas anteriores, o el [borde exterior](#) izquierdo de la caja actual debe estar a la derecha del [borde exterior](#) derecho de la caja anterior, o su parte superior debe estar más abajo que la parte inferior de la caja. Reglas análogas se establecen para las cajas flotantes a la derecha.
3. El [borde exterior](#) derecho de una caja flotante a la izquierda no puede estar a la derecha del [borde exterior](#) izquierdo de ninguna caja flotante derecha que se encuentre a su derecha. Reglas análogas se establecen para los elementos flotantes a la derecha.
4. La [parte superior exterior](#) de una caja flotante no puede estar a mayor altura que la parte superior de su [bloque de contención](#). Cuando una flotante aparece entre dos márgenes que se cierran, la flotante es posicionada como si no hubiera un vacío en el [bloque padre anónimo](#) tomando parte en el flujo. La posición de dicho padre es definida por [las reglas](#) en la sección sobre márgenes cerrados.
5. La [parte superior externa](#) de una caja flotante no puede estar a mayor altura que el borde exterior superior de ninguna caja de [bloque](#) o [flotante](#) generada por un elemento que aparece antes en el documento fuente.
6. La [parte superior externa](#) de la caja flotante de un elemento no puede estar a mayor altura que la parte superior de ninguna [caja en línea](#) que contenga una caja generada por el elemento que aparece antes en el documento fuente.
7. Una caja flotante a la izquierda que tiene otra caja flotante a la izquierda a su izquierda no puede tener su borde exterior derecho a la derecha del borde derecho de su bloque de contención. (en síntesis: una flotantes izquierda no puede estar pegada al borde derecho a menos que ya se encuentra tan a la izquierda como le sea posible). Una regla análoga se sostiene para los elementos flotantes a la derecha.
8. Una caja flotante debe colocarse tan alta como sea posible.
9. Una caja flotante a la izquierda debe ponerse tan a la izquierda como le sea posible, una caja flotante a la derecha tan a la derecha como le sea posible. Una posición mas alta es preferible a una que se encuentre más a la izquierda/derecha.

Referencias para otros elementos en estas reglas se refieren solo a otros elementos en [contexto de formato de bloque](#) como flotante.

9.5.2 Control del flujo al lado de flotantes: la propiedad **'clear'**

'clear'

<i>Valor:</i>	none left right both <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Elementos a nivel de bloque
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

Esta propiedad indica cual de los lados de la(s) caja(s) de un elemento no pueden quedar adyacentes a una caja flotante anterior. La propiedad 'clear' no considera flotantes a los elementos dentro de la misma o en otro [contextos de formato de bloque](#).

Para [cajas run-in](#), esta propiedad se aplica sobre la caja de bloque final a la que la caja run-in pertenece.

El espacio libre (clearance) es introducido como un espacio encima del margen superior de un elemento. Es utilizado para empujar el elemento verticalmente (típicamente hacia abajo), pasada la flotante.

Los valores tienen los siguientes significados cuando se aplican a cajas de bloque no flotantes:

left

El espacio libre (clearance) de una caja generada es aumentado al tamaño necesario para colocar el límite del borde superior por debajo del borde exterior inferior de cualquier caja flotante a la izquierda que sea el resultado de elementos que aparecen antes en el documento fuente.

right

El espacio libre (clearance) de una caja generada se aumenta lo suficiente para colocar el límite del borde superior por debajo del borde exterior inferior de una caja flotante a la derecha que sea el resultado de elementos que aparecen antes en el código fuente del documento.

both

El espacio libre (clearance) de una caja generada se aumento lo suficiente para colocar el límite del borde superior por debajo del borde exterior inferior de una caja flotante a la derecha y una caja flotante a la izquierda que sea resultado de elementos que aparecen antes en el documento fuente.

none

Ninguna restricción a la posición de la caja con respecto a las flotantes.

El calculo del espacio libre (clearance) de un elemento con 'clear' está hecho por la primera determinación de la posición hipotética del límite del borde superior de un elemento dentro de su bloque padre. Esta posición es determinada después de que el margen superior del elemento ha sido cerrado con los márgenes adyacentes previos (incluido el margen superior del bloque padre).

Si el límite del borde superior del elemento no ha pasado las flotantes relevantes, entonces su espacio libre (clearance) es establecido al tamaño necesario para situar el límite del borde del bloque con el límite exterior inferior de la flotante más bajo que puede ser liberado.

Cuando la propiedad es establecida sobre elementos flotantes, se produce una modificación en las [reglas](#) para el posicionamiento de flotantes. Un restricción extra (#10) es añadida:

- El [borde exterior](#) superior de la flotante debe quedar debajo del borde exterior inferior de todas las cajas anteriores flotantes a la izquierda (en el caso de 'clear: left'), o de todas las cajas anteriores flotantes a la derecha (en el caso de 'clear: right') o de ambas ('clear: both').

Nota. Esta propiedad se aplicaba a todos los elementos en CSS1. Las aplicaciones pueden aplicar esta propiedad sobre todos los elementos. En CSS2 y CSS2.1 la propiedad 'clear' solo se aplica a los elementos a nivel de bloque. Por lo tanto los autores deberían solamente utilizar esta propiedad sobre elementos a nivel de bloque. Si una aplicación soporta clear sobre elementos en línea, fijando un espacio libre como se ha explicado arriba, la aplicación debería forzar el salto de línea e insertar efectivamente una o mas cajas en líneas vacías (o desplazar la nueva caja en línea hacia abajo como se describe en la sección 9.5) para mover la parte superior del espacio libre de las líneas de cajas en líneas por debajo de las respectivas cajas flotantes.

9.6 Posicionamiento absoluto

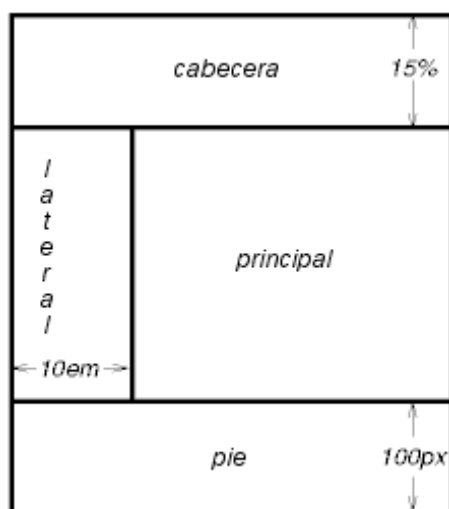
En el modelo de posicionamiento absoluto, una caja es explícitamente desplazada con respecto a su bloque de contención. Es quitada completamente del flujo normal (no tiene ningún impacto sobre los hermanos siguientes). Una caja absolutamente posicionada establece un nuevo bloque de contención para los hijos en el flujo normal y los descendientes posicionados. Sin embargo, el contenido de un elemento posicionado absolutamente no fluye alrededor de ninguna otra caja. Ellos pueden o no tapar el contenido de otra caja, dependiendo del nivel de apilamiento de las cajas solapadas.

Las referencias en esta especificación a un elemento posicionado absolutamente (o su caja) implica que la propiedad 'position' del elemento tiene el valor 'absolute' o 'fixed'.

9.6.1 Posicionamiento fijo (Fixed)

El posicionamiento fijo (fixed) es una subcategoría del posicionamiento absoluto. La única diferencia es que para una caja posicionada fija (fixed), el bloque de contención es establecido por el acceso visual. Para los medios continuos, las cajas fijas (fixed) no se mueven cuando el documento es desplazado. En este aspecto, son similares a las imágenes de fondo fijas. Para los medios paginados, las cajas con posiciones fijas (fixed) son repetidas en cada pagina. Esto resulta útil para poner, por ejemplo, una firma al pie de cada página.

Los autores pueden usar el posicionamiento fijo (fixed) para crear presentaciones al estilo de marcos. Considere la siguiente composición con marcos (frames):



[D]

Esto podría lograrse con el siguiente documento HTML y las reglas de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Un documento con marcos con CSS2.1</TITLE>
    <STYLE type="text/css">
      BODY { height: 8.5in } /* Requerido para los porcentajes de alturas
utilizados abajo */
      #header {
        position: fixed;
        width: 100%;
        height: 15%;
        top: 0;
        right: 0;
        bottom: auto;
        left: 0;
      }
      #sidebar {
        position: fixed;
        width: 10em;
        height: auto;
        top: 15%;
        right: auto;
        bottom: 100px;
        left: 0;
      }
      #main {
        position: fixed;
        width: auto;
        height: auto;
        top: 15%;
        right: 0;
        bottom: 100px;
        left: 10em;
      }
      #footer {
        position: fixed;
        width: 100%;
        height: 100px;
        top: auto;
        right: 0;
        bottom: 0;
        left: 0;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <DIV id="cabecera"> ... </DIV>
    <DIV id="lateral"> ... </DIV>
    <DIV id="principal"> ... </DIV>
    <DIV id="pie"> ... </DIV>
  </BODY>
</HTML>
```

9.7 Relaciones entre 'display', 'position', y 'float'

Las tres propiedades que afectan a la generación de cajas y a la composición — 'display', 'position', y 'float' — interactúan como sigue:

1. Si 'display' tiene el valor 'none', entonces 'position' y 'float' no se aplican. En este caso el elemento no genera ninguna caja.
2. De otro modo, si 'position' tiene el valor 'absolute' o 'fixed', la caja es posicionada absolutamente, el valor computado de 'float' es 'none', y 'display' es colocado de acuerdo a la tabla de abajo. La posición de la caja será determinada por las propiedades 'top', 'right', 'bottom' y 'left' y el bloque de contención de las cajas.
3. De otro modo, si 'float' tiene un valor distinto a 'none', la caja es flotante y 'display' es definida de acuerdo a la tabla de abajo.
4. De otro modo, si el elemento es el elemento raíz, 'display' es colocada de acuerdo a la tabla de abajo.
5. De otro modo, los restantes valores de las propiedades de 'display' se aplican como están especificadas.

Valor especificado	Valor computado
Tabla en línea (inline-table)	table
inline, run-in, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, table-caption, inline-block	block
otros	Igual como el especificado

9.8 Comparación entre flujo normal, flotantes y posicionamiento absoluto

Para ilustrar las diferencias entre flujo normal, posicionamiento relativo, flotantes y posicionamiento absoluto, proporcionaremos una serie de ejemplos basados sobre el siguiente HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Comparación entre esquemas de posicionamientos </TITLE>
  </HEAD>
  <BODY>
    <P>Principio del contenido de body.
      <SPAN id="externo"> Principio del contenido externo.
      <SPAN id="interno"> Contenido interior.</SPAN>
      Fin del contenido externo.</SPAN>
      Fin del contenido de body.
    </P>
  </BODY>
</HTML>
```

En este documento, asumimos las siguientes reglas:

```
body { display: block; font-size:12px; line-height: 200%;  
      width: 400px; height: 400px }  
p    { display: block }  
span { display: inline }
```

La posición final de las cajas generadas por los elementos externos e internos varía en cada ejemplo. En cada ilustración, los números a la izquierda de la ilustración indican la posición del [flujo normal](#) de las líneas con doble espacio (para mayor claridad).

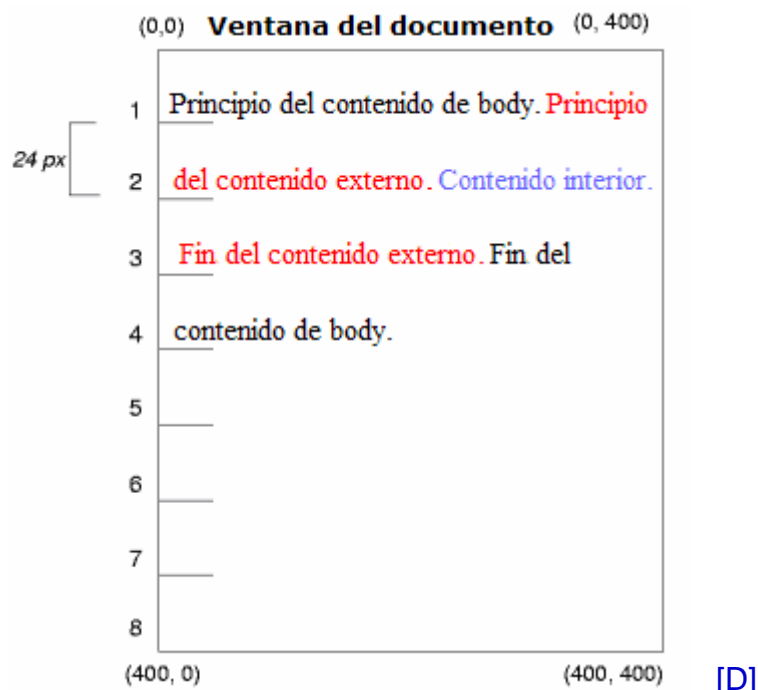
Nota. Los diagramas en esta sección son ilustrativos y no están a escala.. Ellos son propuestos para destacar las diferencias entre los distintos esquemas de posicionamientos en CSS2.1, y no intentan ser referencias del resultado procesado de los ejemplos dados

9.8.1 Flujo Normal

Considere las siguientes declaraciones CSS para externo e interno que no alteran el [flujo normal](#) de las cajas:

```
#externo { color: red }  
#interno { color: blue }
```

El elemento P contiene todo el contenido en línea: [texto en línea anónimo](#) y dos elementos SPAN. Por lo tanto, todo el contenido será colocado en un contexto de formato en línea, con un bloque de contención establecido por el elemento P, produciendo algo similar a:



9.8.2 Posicionamiento relativo

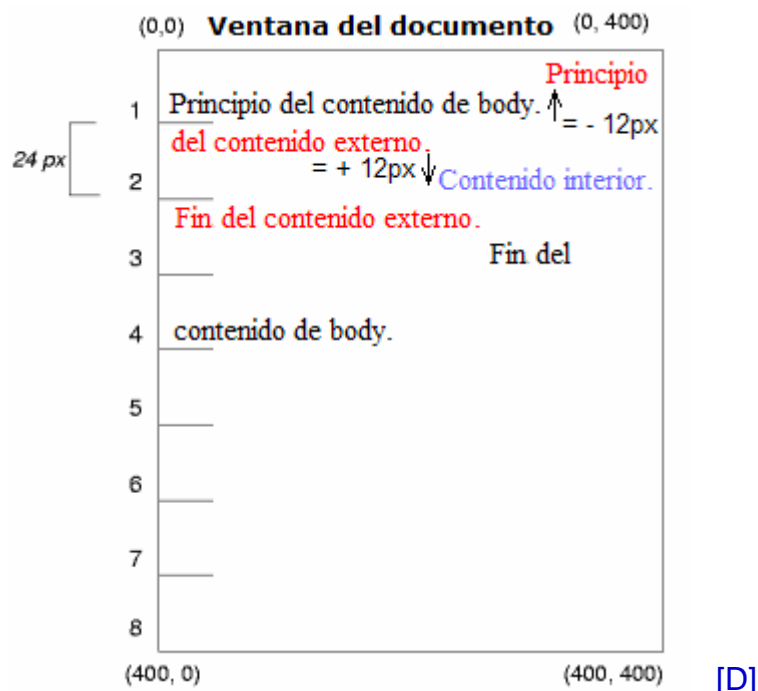
Para ver el efecto del [posicionamiento relativo](#), especificamos:

```
#externo { position: relative; top: -12px; color: red }
#interno { position: relative; top: 12px; color: blue }
```

El texto fluye normalmente hasta el elemento *externo*. El texto *externo* fluye entonces dentro de la posición normal y dimensiones al final de la línea 1. Luego las cajas en línea que contienen el texto (distribuido en tres líneas) son movidas como una unidad en '-12px' (hacia arriba).

El contenido interno, como un hijo del externo, debería fluir normalmente inmediatamente después de las palabras "del contenido externo" (en la línea 1.5). Sin embargo, el propio contenido interno está desplazada con respecto al contenido *externo* en '12px' (hacia abajo), regresando a su posición original en la línea 2.

Observe que el contenido que sigue a *externo* no es afectado por el posicionamiento relativo de *externo*.



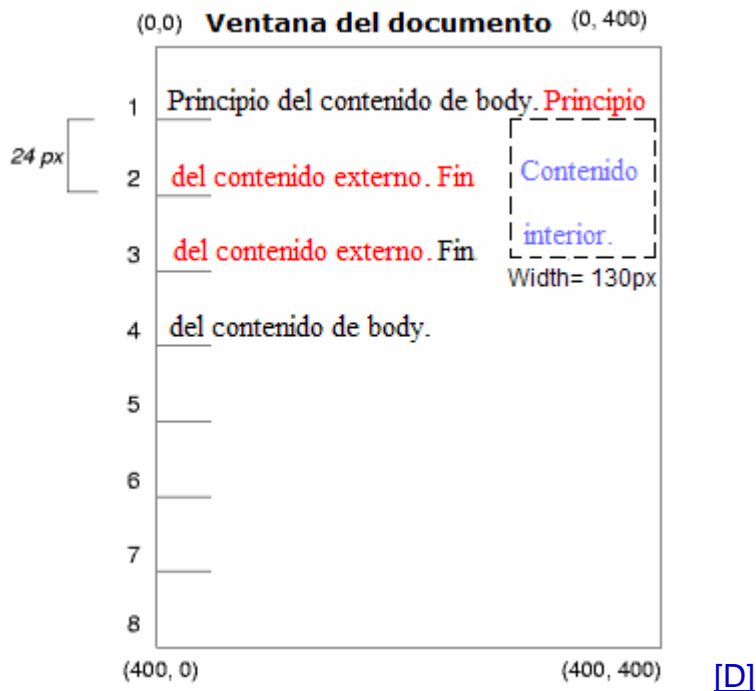
Observe que si el desplazamiento de *externo* hubiera sido de '-24px', el texto de *externo* y el texto del contenido se habrían solapado.

9.8.3 Flotar una caja

Ahora considere el efecto de [flotar](#) el texto del elemento *interno* a la derecha por medio de las siguientes reglas:

```
#externo { color: red }
#interno { float: right; width: 130px; color: blue }
```

El texto fluye normalmente hasta la caja de *interno*, la que es empujada fuera del flujo y flotada sobre el margen derecho (su propiedad `'width'` ha sido asignada explícitamente). Las líneas de las cajas a la izquierda de la flotante son cortadas, y el texto del documento restante fluye dentro de ellas.



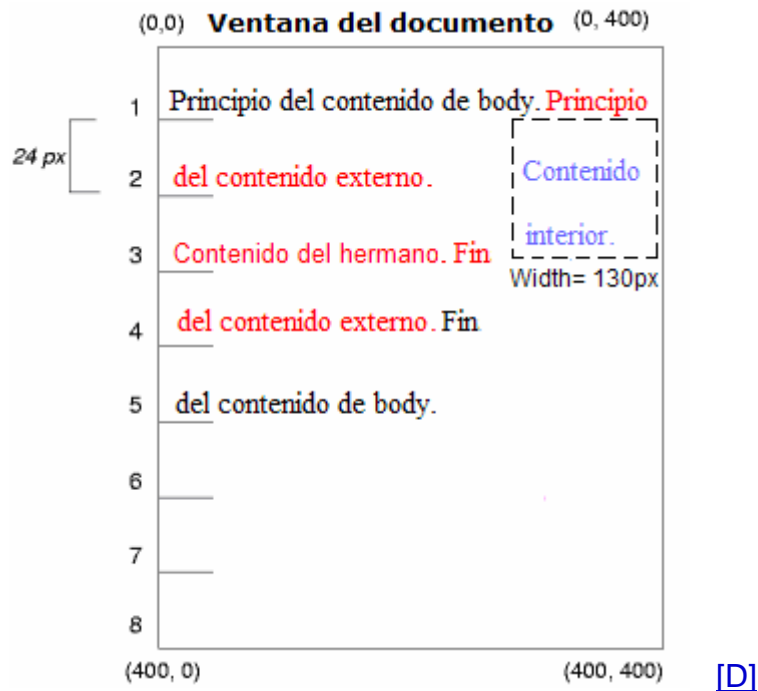
Para mostrar el efecto de la propiedad `'clear'`, agregaremos un elemento hermano al ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Comparison of positioning schemes II</TITLE>
  </HEAD>
  <BODY>
    <P>Principio del contenido de body.
      <SPAN id=externo> Principio del contenido externo.
      <SPAN id=interno> Contenido interno.</SPAN>
      <SPAN id=hermano> Contenido del hermano.</SPAN>
      Fin del contenido externo.</SPAN>
      Fin del contenido de body.
    </P>
  </BODY>
</HTML>
```

Las siguiente reglas:

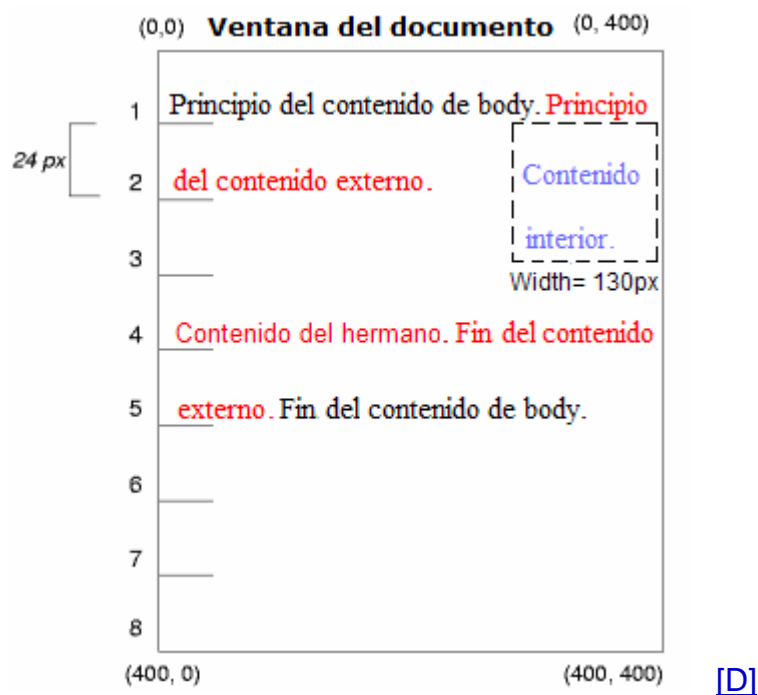
```
#interno { float: right; width: 130px; color: blue }
#hermano { color: red }
```

Provoca que la caja de interno flote a la derecha como antes y el texto del documento restantes fluye dentro del espacio vacante:



Sin embargo, si la propiedad `clear` en el elemento *hermano* es puesta como 'right' (es decir, la caja hermana generada no aceptará una posición al lado de las cajas flotantes a su derecha), el contenido del hermano comienza a fluir debajo del flotante:

```
#interno { float: right; width: 130px; color: blue }
#hermano { clear: right; color: red }
```

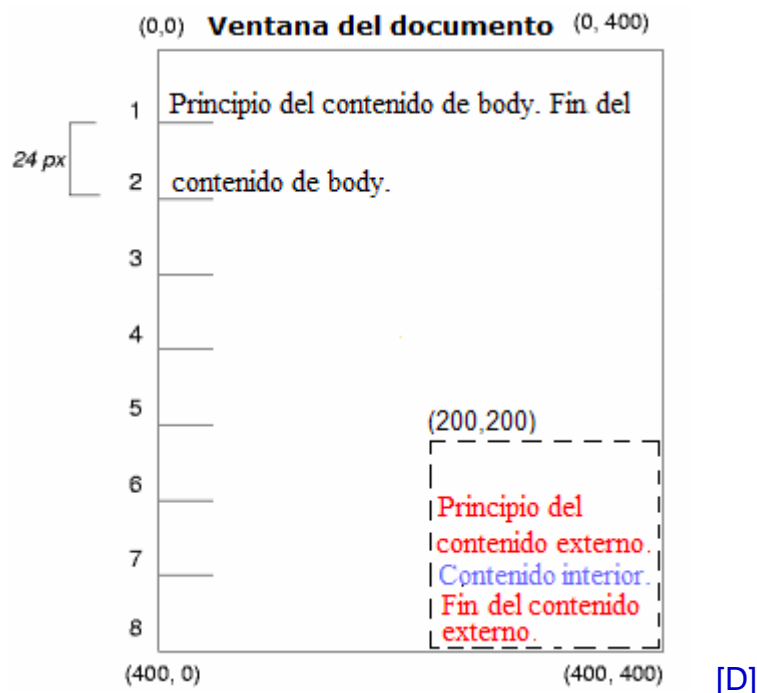


9.8.4 Posicionamiento absoluto

Finalmente, consideramos el efecto del [posicionamiento absoluto](#). Considere las siguientes declaraciones CSS para *externo* e *interno*:

```
#externo {  
    position: absolute;  
    top: 200px; left: 200px;  
    width: 200px;  
    color: red;  
}  
#interno { color: blue }
```

Que provocan que el lado superior de la caja de *externo* sea posicionada con respecto a su bloque de contención. El bloque de contención para una caja posicionada se establece por el antepasado posicionado más cercano (o, si no existe, el [bloque de contención inicial](#), como en nuestro ejemplo). El lado superior de la caja *externo* está '200px' debajo de la parte superior del bloque de contención y el lado izquierdo está a '200px' del lado izquierdo. La caja hija de *externo* fluye normalmente con respecto a su padre.



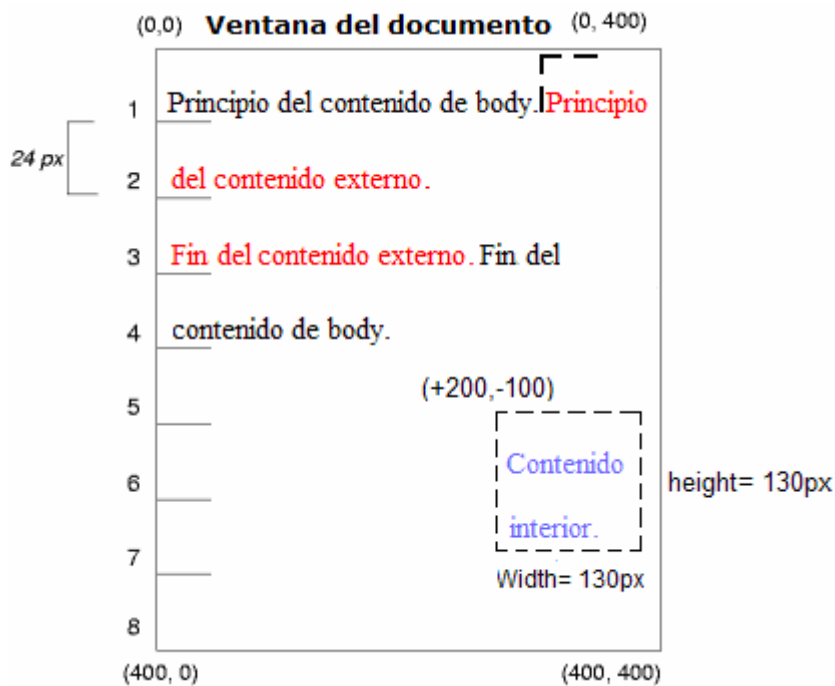
El siguiente ejemplo muestra una caja posicionada absolutamente que es hija de una caja posicionada relativamente. Aunque la caja padre *externo* no está realmente desplazada, poniendo su propiedad ['position'](#) como 'relative' significa que su caja puede servir como bloque de contención para los descendientes posicionados. Como la caja *externo* es una caja en línea es dividida entre varias líneas, los bordes superior e izquierdo de la primera caja en línea (señaladas con líneas cortadas gruesas en la ilustración de abajo) sirven como referencia para los desplazamientos ['top'](#) and ['left'](#).

```

#externo {
  position: relative;
  color: red
}
#interno {
  position: absolute;
  top: 200px; left: -100px;
  height: 130px; width: 130px;
  color: blue;
}

```

Esto da como resultado algo similar a lo siguiente:



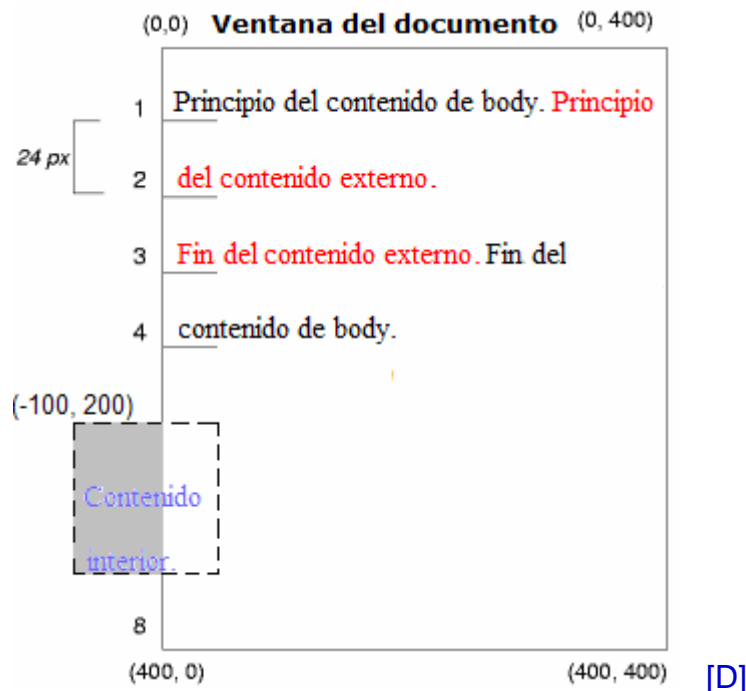
Si no posicionamos la caja de externo:

```

#externo { color: red }
#interno {
  position: absolute;
  top: 200px; left: -100px;
  height: 130px; width: 130px;
  color: blue;
}

```

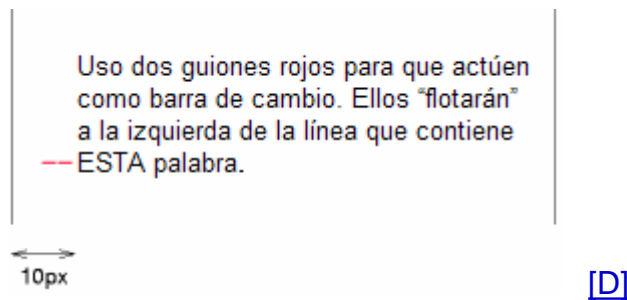
El bloque de contención para *interno* es ahora el [bloque de contención inicial](#) (en nuestro ejemplo). La siguiente ilustración muestra donde acabaría la caja de interno en este caso.



El posicionamiento relativo y absoluto puede utilizarse para implementar barras de cambio, como se muestra en el siguiente ejemplo. El fragmento siguiente:

```
<P style="position: relative; margin-right: 10px; left: 10px;">
  Uso dos guiones rojos para que actúen como barra de cambio.
  Ellos "flotarán" a la izquierda de la línea que contiene ESTA
  <SPAN style="position: absolute; top: auto; left: -1em; color: red;">--
  </SPAN>
  palabra.</P>
```

resultaría algo como:



Primero, el párrafo (cuyos lados del bloque de contención son mostrados en la ilustración) fluye normalmente. Entonces es desplazado '10px' desde el borde izquierdo de su bloque de contención (así también, un margen derecho de '10px' ha sido reservado en anticipación al desplazamiento). Los dos guiones que actúan como barras de cambio son sacadas del flujo y posicionadas en la línea actual (debido a 'top: auto'), '-1em' desde el margen izquierdo de su bloque de contención (establecido por P en su posición final). El resultado es que la barra de cambio parece "flotar" a la izquierda de la línea actual.

9.9 Presentación por capas

9.9.1 Especificar el nivel de apilamiento: la propiedad **'z-index'**

'z-index'

Valor:	auto <u><entero></u> <u>inherit</u>
Inicial:	auto
Se aplica a:	Elementos posicionados
Se hereda:	no
Porcentajes:	N/A
Medio:	<u>visual</u>
Valor computado:	Como el especificado

Para una caja posicionada, la propiedad 'z-index' especifica:

1. El nivel de apilamiento (pila) de la caja en el contexto de apilamiento (pila) actual
2. Si la caja establece un contexto de apilamiento local.

Los valores tienen los siguientes significados:

<entero>

Este entero es el nivel de pila de la caja generada en el contexto del apilamiento actual. La caja también establece un contexto de apilamiento local en el cual su nivel de pila es '0'.

auto

El nivel de pila de la caja generada en el contexto de apilamiento actual es el mismo que el de la caja padre. La caja no establece un nuevo contexto de apilamiento local.

En esta sección, la expresión "en frente de" significa mas cercano al usuario según el usuario situado frente a la pantalla.

En CSS2.1, cada caja tiene una posición en tres dimensiones. Además de su posición horizontal y vertical, las cajas se ubican a lo largo de un eje "z" y son procesadas una sobre la otra. Las posiciones Z son particularmente relevantes cuando las cajas se superponen visualmente. Esta sección trata de cómo las cajas pueden ser posicionadas a lo largo del eje z.

El orden en el que la estructura (del documento) procesada es pintada sobre el lienzo es descrito en términos del contexto de apilamiento. El contexto de apilamiento puede contener apoyo a otros contextos de apilamiento. Un contexto de apilamiento es atómico desde el punto de vista del contexto de apilamiento de su padre; las cajas en otros contextos de apilamiento no pueden aparecer entre ninguna de sus cajas.

Cada caja pertenece a un contexto de apilamiento. Cada caja en un contexto de apilamiento dado tiene un entero como nivel de pila, el cuál es su posición en el eje z en relación a otras cajas en el mismo contexto de apilamiento. Las cajas creadas con niveles de pila mayores son siempre procesados en frente (encima) de las cajas con niveles de pila menores. Las cajas pueden tener niveles de pila negativos. Las cajas con el mismo nivel de pila en un contexto de apilamiento son apiladas de abajo hacia arriba de acuerdo al orden en la estructura del documento

El elemento raíz crea un contexto de apilamiento raíz. Otros contextos de apilamiento son generados por algún elemento posicionado (incluido los elementos posicionados relativamente) que tengan un valor computado de 'z-index' distinto de 'auto'. Los contextos de apilamiento no están necesariamente relacionados con bloques de contención. En futuros niveles de CSS, otras propiedades pueden introducir contextos de apilamiento, por ejemplo ['opacity'](#).

Cada contexto de apilamiento consiste en los siguientes niveles de apilamiento (pila) (de atrás a delante):

1. El fondo y los bordes del elemento que forman el contexto de apilamiento.
2. El contexto de apilamiento de descendentes con niveles de pila negativos.
3. Un nivel de pila que contiene descendentes no a nivel de línea en flujo.
4. Un nivel de pila para flotantes y su contenido.
5. Un nivel de pila para descendentes a nivel de línea en flujo.
6. Un nivel de pila para descendentes posicionados con 'z-index: auto', y algún contexto de apilamiento descendente con 'z-index: 0'.
7. El contexto de apilamiento de descendentes con niveles de pila positivas.

El contenido de bloques en línea y tablas en línea son apilados como si ellos generasen nuevos contextos de apilamiento, excepto algunos elementos que en realidad crean nuevos contextos de apilamiento que toman parte en el contexto de apilamiento del padre. Estos son pintados automáticamente en el nivel de apilamiento en línea.

En el siguiente ejemplo, los niveles de pila de las cajas (nombrados con sus atributos "id") son: "text2"=0, "image"=1, "text3"=2, y "text1"=3. El nivel de pila de "text2" es heredado de la caja raíz. Los otros son especificados con la propiedad ['z-index'](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Z-order positioning</TITLE>
    <STYLE type="text/css">
      .pila {
        position: absolute;
        left: 2in;
        top: 2in;
        width: 3in;
        height: 3in;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P>
      <IMG id="image" class="pila"
        src="mariposa.png" alt="La imagen de una mariposa"
        style="z-index: 1">

      <DIV id="text1" class="pila"
        style="z-index: 3">
        Este texto quedará sobre la imagen de la mariposa.
      </DIV>
```

```
<DIV id="text2">
  Este texto quedará detrás de todo.
</DIV>

<DIV id="text3" class="pile"
  style="z-index: 2">
  Este texto quedará debajo del text1, pero sobre la imagen de la
mariposa.
</DIV>
</BODY>
</HTML>
```

Este ejemplo demuestra la noción de transparencia. El comportamiento predeterminado de una caja es permitir que las cajas que quedan detrás sean visibles a través de las áreas transparentes en su contenido. En el ejemplo, cada caja se superpone transparentemente con las cajas debajo de ellas. Este comportamiento puede ser anulado usando una las [propiedades del fondo](#).

9.10 Dirección del texto: las propiedades 'direction' y 'unicode-bidi'

Las aplicaciones de usuarios con [conformidad](#) que no soporten los textos bidireccionales pueden ignorar las propiedades 'direction' y 'unicode-bidi' descritas en esta sección.

Los caracteres en ciertas escrituras se escriben de derecha a izquierda. En algunos documentos, en particular aquellos escritos en Árabe o en Hebreo, y en algunos contextos de lenguaje mixto, el texto de un único (mostrado visualmente) bloque puede aparecer con direccionalidad mixta. Este fenómeno se llama bidireccionalidad, o "bidi" para abreviar.

El estándar Unicode ([\[UNICODE\]](#), sección 3.11) define un complejo algoritmo para determinar la direccionalidad del texto. El algoritmo se compone de una parte implícita basada en las propiedades de caracteres, así como de controles explícitos para las incrustaciones (embebidos) y sustituciones. CSS2.1 confía en este algoritmo para lograr un adecuado procesamiento bidireccional. Las propiedades 'direction' y 'unicode-bidi' permiten a los autores especificar como los elementos y atributos del lenguaje de un documento se vinculan con este algoritmo.

Si un documento contiene caracteres de derecha a izquierda, y si la aplicación del usuario muestra esos caracteres en orden de derecha a izquierda, la aplicación del usuario puede aplicar el algoritmo bidireccional (Aun que procesan caracteres simples de derecha a izquierda porque una fuente del sistema las contiene pero no soporta el concepto de dirección del texto de derecha a izquierda está exenta de este requisito.)

Debido a que la direccionalidad de un texto depende de la estructura y semántica del lenguaje del documento, estas propiedades deben en la mayoría de los casos ser usadas por los diseñadores de descripciones del tipo de documento (DTD), o autores de documentos especiales. Si una hoja de estilo predeterminada especifica estas propiedades, los autores y usuarios no deben especificar reglas para sustituirlas.

La especificación HTML 4.0 ([HTML40], sección 8.2) define el comportamiento de la bidireccionalidad para elementos HTML. Las reglas de las hojas de estilo que puedan alcanzar el comportamiento bidi especificado en [\[HTML40\]](#) son dadas en [hoja de estilo de ejemplo](#). La especificación HTML 4.0 contiene mas información sobre cuestiones de bidireccionalidad.

'direction'

Valor: ltr | rtl | inherit
Inicial: ltr
Se aplica a: Todos los elementos, pero ver el texto
Se hereda: Si
Porcentajes: N/A
Medio: visual
Valor computado: Como el especificado

Esta propiedad especifica la dirección básica de escritura de los bloques y la dirección de las incrustaciones y sustituciones (ver 'unicode-bidi') para el algoritmo bidireccional de Unicode. Además, especifica la dirección de la composición de las columnas en una [tabla](#), la dirección del [desbordamiento](#) horizontal, y la posición de la ultima línea incompleta en un bloque en el caso de 'text-align: justify'.

Los valores para esta propiedad tiene los siguientes significados:

ltr Dirección de izquierda a derecha (Left-to-right).
rtl Dirección de derecha a izquierda (Right-to-left).

Para que la propiedad 'direction' afecte al reordenamiento en elementos a nivel de línea, el valor de la propiedad 'unicode-bidi' debe ser 'embed' o 'override'.

***Nota.** La propiedad 'direction', cuando se especifica para columnas de tablas, no es heredado por las celdas en la columna ya que las columnas no son los antepasados de las celdas en la estructura del documento. Por ello, CSS no puede capturar fácilmente las reglas de herencia del atributo "dir" descritas en [\[HTML40\]](#), sección 11.3.2.1.*

'unicode-bidi'

Valor: normal | embed | bidi-override | inherit
Inicial: normal
Se aplica a: Todos los elementos, pero ver el texto
Se hereda: no
Porcentajes: N/A
Medio: visual
Valor computado: Como el especificado

Los valores para esta propiedad tiene los siguientes significados:

normal

El elemento no abre un nivel adicional de incrustación con respecto al algoritmo bidireccional. Para los elementos a nivel de línea, el reordenamiento implícito funciona a través de todos los límites del elemento.

embed

Si el elemento es a nivel de línea, este valor abre un nivel adicional de incrustación con respecto al algoritmo bidireccional. La dirección de este nivel de incrustación está dada por la propiedad 'direction'. Dentro del elemento, el reordenamiento está hecho implícitamente. Esto corresponde a agregar un LRE (U+202A; para 'direction: ltr') o RLE (U+202B; para 'direction: rtl') al comienzo del elemento y un PDF (U+202C) al final del elemento.

bid-override

Para elementos a nivel de línea esto crea una sustitución. Para elementos a nivel de bloque (block-level), table-cell (celda de tabla), table-caption (titulo de tabla), o bloques en línea (inline-block), esto crea una sustitución para el descendente a nivel de línea no dentro de otros elementos a nivel de bloque (block-level), table-cell (celda de tabla), table-caption (titulo de tabla), o bloques en línea (inline-block). Esto significa que dentro del elemento, el reordenamiento es estrictamente en secuencia según la propiedad 'direction'; la parte implícita del algoritmo bidireccional es ignorada. Esto corresponde a agregar un LRO (U+202D; para 'direction: ltr') o RLO (U+202E; para 'direction: rtl') al comienzo del elemento y un PDF (U+202C) al final del elemento.

El orden final de los caracteres en cada elemento a nivel de bloque es el mismo que si los códigos de control bidi se hubieran agregado como se describe arriba, las marcas se hubieran quitado y la secuencia de caracteres resultante hubiera pasado a una implementación del algoritmo bidireccional de Unicode para un texto puro que produce los mismos saltos de línea que el texto con estilo. En este proceso, las entidades no textuales como las imágenes son tratadas como caracteres neutros, a menos que su propiedad 'unicode-bidi' tenga un valor distinto a 'normal', en cuyo caso son tratadas como caracteres gruesos en la 'direction' especificada para el elemento.

Por favor observe que para conseguir un flujo de las cajas en línea con una dirección uniforme (ya sea completamente de izquierda a derecha o completamente de derecha a izquierda), pueden crearse mas cajas en línea (incluyendo cajas en línea anónimas), y algunas cajas en línea pueden tener que ser divididas y reordenadas antes de fluir.

Debido a que el algoritmo Unicode tiene un límite de 61 niveles de incrustación, se debe cuidar de no usar 'unicode-bidi' con un valor distinto a 'normal' a menos que sea apropiado. En particular, un valor 'inherit' debería usarse con extrema precaución. Sin embargo, para los elementos que son, en general, pensados para ser mostrados como bloques, se prefiere la definición de 'unicode-bidi: embed' para mantener el elemento junto en caso de que la visualización cambie a nivel de línea (ver el ejemplo de abajo).

El siguiente ejemplo muestra un documento XML con texto bidireccional. Ilustra un importante principio de diseño: los diseñadores de DTD deben tomar en cuenta la bidireccionalidad en el propio lenguaje (elementos y atributos) y en cualquier hoja de estilo que acompañe. Las hojas de estilo deben ser diseñadas de manera tal que las reglas bidi queden separadas de las otras reglas de estilo. Las reglas bidi no deben ser sustituidas por otras hojas de estilo de modo que el comportamiento de la bidireccionalidad del lenguaje del documento o la DTD sea preservada.

En este ejemplo, las letras en minúsculas se usan para los caracteres de izquierda a derecha heredados y las mayúsculas representan los caracteres de derecha a izquierda heredados:


```

<HEBREO>
  <PAR>HEBBREO1 HEBREO2 ingles3 HEBREO4 HEBREO5</PAR>
  <PAR>HEBREO6 <EMPH>HEBREO7</EMPH> HEBREO8</PAR>
</HEBREO>
<INGLES>
  <PAR>ingles9 ingles10 ingles11 HEBREO12 HEBREO13</PAR>
  <PAR> ingles14 ingles15 ingles16</PAR>
  <PAR> ingles17 <HE-CITAQUO>HEBREO18 ingles19 HEBREO20</HE-CITAQUO></PAR>
</INGLES>

```

Como esto es XML, la hoja de estilo es la responsable de definir la dirección de escritura. Esta es la hoja de estilo:

```

/* Reglas para bidi */
HEBREO, HE-CITA {direction: rtl; unicode-bidi: embed}
INGLES          {direction: ltr; unicode-bidi: embed}

/* Reglas para presentación */
HEBREO, INGLES, PAR {display: block}
EMPH              {font-weight: bold}

```

El elemento HEBREO es un bloque con una dirección básica de derecha a izquierda, el elemento INGLES es un bloque con una dirección básica de izquierda a derecha. Los PARs son bloques que heredan la dirección básica de sus padres. Así, los primeros dos PARs se leen comenzando desde arriba a la derecha, los tres finales se leen comenzando desde arriba a la izquierda. Por favor, observe que HEBREO e INGLES son elegidos como nombres de elementos solamente por ser mas explícitos, los nombres de los elementos pueden proporcionar la estructura sin referencia al idioma.

El elemento EMPH es a nivel de línea, y como su valor para 'unicode-bidi' es 'normal' (el valor inicial), no tiene ningún efecto en el ordenamiento del texto. El elemento HE-CITA, por otra parte, crea una incrustación.

El formato de este texto podría parecerse a esto si la línea es larga:

```

      5OERBEH 4OERBEH ingles3 2OERBEH 1OERBEH
                8OERBEH 7OERBEH 6OERBEH

ingles9 ingles10 ingles11 13OERBEH 12OERBEH

ingles14 ingles15 ingles16

ingles17 20OERBEH ingles19 18OERBEH

```

Observe que la incrustación HE-CITA provoca que HEBREO18 quede a la derecha de ingles19.

Si las líneas tiene que ser cortadas, sería más como esto:

```

      2OERBEH 1OERBEH
-EH 4OERBEH english3
      5OERB

```

-EH 7OERBEH 6OERBEH
8OERB

ingles9 ingles10 in-
gles11 12OERBEH
13OERBEH

ingles14 ingles15
ingles16

ingles17 18OERBEH
20OERBEH ingles19

Como HEBREO18 debe leerse antes que ingles19, está en la línea encima de español19. Sólo cortando la línea larga de formato anterior no hubiera funcionado. Observe también que la primera sílaba de ingles19 podría caber en la línea anterior, pero el silabeo de las palabras de izquierda a derecha en un contexto de derecha a izquierda, y viceversa, es normalmente suprimido para evitar tener de mostrar un guión en el medio de la línea.

10 Detalles del modelo de formato visual

Contenido

- [10.1 Definición de "bloque de contención"](#)
- [10.2 Ancho del contenido: la propiedad 'width'](#)
- [10.3 Computando anchos y márgenes](#)
 - [10.3.1 Elementos no reemplazados, en línea](#)
 - [10.3.2 Elementos reemplazados, en línea](#)
 - [10.3.3 Elementos no reemplazados en flujo normal, a nivel de bloque](#)
 - [10.3.4 Elementos reemplazados en flujo normal, a nivel de bloque](#)
 - [10.3.5 Elementos no reemplazados, flotantes](#)
 - [10.3.6 Elementos reemplazados, flotantes](#)
 - [10.3.7 Elementos no reemplazados, posicionados absolutamente](#)
 - [10.3.8 Elementos reemplazados, posicionados absolutamente](#)
 - [10.3.9 Elementos no reemplazados en flujo normal, de 'bloques en línea'](#)
 - [10.3.10 Elementos reemplazados en flujo normal, de 'bloques en línea'](#)
- [10.4 Anchos mínimos y máximos: 'min-width' y 'max-width'](#)
- [10.5 Altura del contenido: la propiedad 'height'](#)
- [10.6 Computando alturas y márgenes](#)
 - [10.6.1 Elementos no reemplazados, en línea](#)
 - [10.6.2 Elementos reemplazados, en línea, elementos reemplazados en flujo normal a nivel de bloque, elementos reemplazados en flujo normal de bloques en línea y elementos reemplazados flotantes](#)
 - [10.6.3 Elementos no reemplazados en flujo normal a nivel de bloque cuando 'overflow' es computado como 'visible'](#)
 - [10.6.4 Elementos no reemplazados, posicionados absolutamente](#)
 - [10.6.5 Elementos reemplazados, posicionados absolutamente](#)
 - [10.6.6 Elementos no reemplazados en flujo normal a nivel de bloque, cuando 'overflow' no es computado como 'visible', elementos no reemplazados de bloques en línea, y elementos no reemplazados flotantes](#)
 - [10.6.7 Alturas 'auto' para el contexto de formato de bloque raíz](#)
- [10.7 Alturas mínimas y máximas: 'min-height' y 'max-height'](#)
- [10.8 Cálculo de la altura de la línea: las propiedades 'line-height' y 'vertical-height'](#)
 - [10.8.1 Interlineado y medio interlineado](#)

10.1 Definición de "bloque de contención"

La posición y tamaño de la(s) caja(s) de un elemento a veces son computadas en relación a cierto rectángulo, llamado el *bloque de contención* del elemento. El bloque de contención de un elemento se define como sigue:

1. El bloque de contención en el cual reside el [elemento raíz](#) es elegido por la aplicación del usuario. (podría ser relacionado con el [acceso visual](#)) Este bloque de contención se llama *bloque de contención inicial*.
2. Para otros elementos, si la posición del elemento es 'relative' o 'static', el bloque de contención está formado por el borde del contenido del antepasado más cercano de la [caja a nivel de bloque](#), celda de tabla o cajas en líneas.

3. Si el elemento tiene 'position: fixed', el bloque de contención es establecido por el [acceso visual](#) en el caso de medios continuos o la [caja de la pagina](#) en el caso de medios paginados
4. Si el elemento tiene 'position: absolute', el bloque de contención es establecido por el antepasado más cercano con 'position' 'absolute', 'relative' o 'fixed', de la manera siguiente:
 1. En el caso de que el antepasado sea a nivel de línea, el bloque de contención depende de la propiedad 'direction' del antepasado:
 1. Si 'direction' es 'ltr', la parte superior e izquierda del bloque de contención son la parte superior e izquierda del borde del contenido de la primera caja generada por el antepasado, y la parte inferior y derecha es la parte inferior y derecha del borde del contenido de la última caja del antepasado.
 2. Si 'direction' es 'rtl', la parte superior y derecha son los bordes superior y derecho de la primera caja generada por el antepasado, y la parte inferior e izquierda son el borde inferior e izquierdo del contenido de la última caja del antepasado.
 2. Si no, el bloque de contención es formado por el [borde del relleno](#) del antepasado.

Si no hay tal antepasado, el bloque de contención es el bloque de contención inicial.

En medios paginados, un elemento posicionado absolutamente es posicionado relativamente a su bloque de contención ignorando cualquier salto de pagina (como si el documento fuese continuo). El elemento puede romperse posteriormente en varias páginas.

Observe que un elemento a nivel de bloque que es dividido en varias paginas puede tener diferentes anchos en cada pagina y pueden haber limites específicos de dispositivos.

Si posicionamos, los bloques de contención (B.C.) en el documento siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<HTML>
  <HEAD>
    <TITLE>Ilustración de bloques de contención</TITLE>
  </HEAD>
  <BODY id="body">
    <DIV id="div1">
      <P id="p1">Este es el texto en el primer párrafo...</P>
      <P id="p2">Este es el <EM id="em1"> en el
      <STRONG id="strong1">segundo</STRONG> párrafo.</EM></P>
    </DIV>
  </BODY>
</HTML>
```

se establecen como sigue:

Para la caja generada por	B.C. es establecido por
html	B.C. inicial (depende de la AU)
body	Html
div1	body
p1	div1
p2	div1
em1	p2
strong1	p2

Si posicionamos "div1":

```
#div1 { position: absolute; left: 50px; top: 50px }
```

su bloque de contención no es más "body"; se convierte en el bloque de contención inicial (porque no hay otras cajas antepasadas posicionadas).

Si también posicionamos "em1":

```
#div1 { position: absolute; left: 50px; top: 50px }  
#em1 { position: absolute; left: 100px; top: 100px }
```

la tabla de bloques de contención se convierte en:

Para la caja generada por	B.C. es establecido por
html	B.C. inicial (dependen de la AU)
body	Html
div1	B.C. inicial
p1	div1
p2	div1
em1	div1
strong1	em1

Posicionando "em1", su bloque de contención se convierte en la caja del antepasado posicionado más cercano (es decir, el generado por "div1").

10.2 Ancho del contenido: la propiedad **'width'**

'width'

<i>Valor:</i>	<u><medida></u> <porcentaje> auto inherit
<i>Inicial:</i>	Auto
<i>Se aplica a:</i>	todos los elementos excepto elementos en línea no reemplazados, filas de tabla, y grupos de fila
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	Referidos al ancho del bloque de contención
<i>Medios:</i>	<u>visual</u>
<i>Valor computado:</i>	porcentaje o 'auto' según lo especificado o la medida absoluta; 'auto' si la propiedad no se aplica

Esta propiedad especifica el [ancho del contenido](#) de las cajas generadas por elementos a nivel de bloque y [reemplazados](#).

Esta propiedad no se aplica a los elementos a [nivel de línea](#) no reemplazados. La anchura del contenido de las cajas de elementos en línea no reemplazados es el de su contenido generado (antes de cualquier desplazamiento relativo de los hijos). Recuerde que las cajas en línea fluyen dentro de las [líneas de las cajas](#). La anchura de las cajas en línea son dadas por su [bloque de contención](#), pero puede ser acortado por la presencia de [flotantes](#).

El ancho de la caja de un elemento reemplazado es [intrínseco](#) y puede ser modificado por la aplicación del usuario si el valor de esta propiedad es distinto a 'auto'.

Los valores tienen los significados siguientes:

<medida>

Especifica el ancho del área de contenido usando una unidad de medida.

<porcentaje>

Especifica un anchura según el porcentaje. El porcentaje se calcula con respecto a la anchura del [bloque de contención](#) de las cajas generadas. Si el ancho del bloque de contención depende del ancho de este elemento, entonces el resultado mostrado no es definido en CSS2.1. *Nota: Para los elementos posicionados absolutamente que contienen bloques basados en un elemento a nivel de bloque, el porcentaje se calcula con respecto al ancho del relleno de la caja de ese elemento. Esto es un cambio de CSS1, donde el porcentaje del ancho era calculado siempre con respecto *al contenido de la caja* del elemento padre.*

auto

El ancho depende de los valores de otras propiedades. Vea las secciones de abajo.

Los valores negativos para 'width' son ilícitos.

Por ejemplo, la regla siguiente fija el ancho del contenido de los párrafos en 100 píxeles:

```
p { width: 100px }
```

10.3 Computando anchos y márgenes

Los valores de las propiedades `'width'`, `'margin-left'`, `'margin-right'`, `'left'` y `'right'` de un elemento usados para composición dependen del tipo de caja generada y entre sí. (el valor usado para la composición se refiere a veces como el [valor usado](#)) En principio, los valores usados son iguales que los valores computados, con 'auto' reemplazado por algún valor apropiado, y los porcentajes calculados en base al bloque de contenido, pero hay excepciones. Las siguientes situaciones necesitan distinguirse:

1. Elementos no reemplazados, en línea
2. Elementos reemplazados, en línea
3. Elementos no reemplazados en flujo normal, a nivel de bloque
4. Elementos reemplazados en flujo normal, a nivel de bloque
5. Elementos no reemplazados, flotantes
6. Elementos reemplazados, flotantes
7. Elementos no reemplazados, posicionados absolutamente
8. Elementos reemplazados, posicionados absolutamente
9. Elementos no reemplazados en flujo normal, 'bloque en línea'
10. Elementos reemplazados en flujo normal, 'bloque en línea'

Para los puntos 1-6 y 9-10, los valores de 'left' y 'right' usados para la composición son determinados por las reglas en la [sección 9.4.3](#).

10.3.1 Elementos no reemplazados, en línea

La propiedad `'width'` no se aplica. Un valor computado 'auto' para `'left'`, `'right'`, `'margin-left'` o `'margin-right'` se convierte en un valor usado de '0'.

10.3.2 Elementos reemplazados , en línea,

Un valor computado de 'auto' para `'margin-left'` o `'margin-right'` se convierte en un valor usado de '0'. Si `'width'` tiene un valor computado de 'auto' y `'height'` también tiene un valor computado de 'auto', el ancho `'intrínseco'` del elemento utiliza el valor de `'width'`. Si `'width'` tiene un valor computado de 'auto' y `'height'` tiene otro valor computado, entonces el valor usado de `'width'` es:

$$(\text{ancho intrínseco}) * ((\text{alto usado})/(\text{alto intrínseco}))$$

10.3.3 Elementos no reemplazados en flujo normal, a nivel de bloque

Las siguientes condiciones deben mantenerse entre los valores utilizados entre todas las propiedades:

`'margin-left'` + `'border-left-width'` + `'padding-left'` + `'width'` + `'padding-right'` + `'border-right-width'` + `'margin-right'` = ancho del [bloque de contención](#)

Si `'width'` no es 'auto' y `'border-left-width'` + `'padding-left'` + `'width'` + `'padding-right'` + `'border-right-width'` (mas algún `'margin-left'` o `'margin-right'` que no sean 'auto') es más grande que el ancho del bloque de contención, entonces cualquier valor 'auto' de `'margin-left'` o `'margin-right'` son, para las reglas siguientes, tomadas como cero.

Si todos los anteriores tienen un valor computado distinto de 'auto', los valores serían "sobre-obligados" y uno de los valores usados tendrá que ser diferente de su valor computado. Si la propiedad 'direction' tiene el valor 'ltr', el valor especificado de 'margin-right' es ignorado y el valor es calculado para hacer la igualdad verdadera. Si el valor de 'direction' es 'rtl', esto a su vez sucede con 'margin-left'.

Si hay exactamente un valor especificado como 'auto', su valor computado proviene de la igualdad.

Si 'width' se fija como 'auto', cualquier otro valor 'auto' se convierte en '0' y 'width' proviene de la igualdad resultante.

Si ambos, 'margin-left' y 'margin-right' son 'auto', sus valores usados son iguales. Esto centra horizontalmente el elemento con respecto a los bordes del bloque de contención.

10.3.4 Elementos reemplazados en flujo normal, a nivel de bloque

El valor usado de 'width' se determina igual que para [elementos reemplazados en línea](#). Entonces las reglas [para los elementos no reemplazados a nivel de bloque](#) se aplican para determinar los márgenes.

10.3.5 Elementos no reemplazados, Flotantes

Si 'margin-left', o 'margin-right' se computan como 'auto', su valor usado es '0'.

Si 'width' se computa como 'auto', el valor usado es "encogido al intervalo" (shrink-to-fit) del ancho.

El "encogimiento al intervalo" del ancho se calcula igual que el ancho de una celda de una tabla utilizando el algoritmo automático de representación de la tabla. Aproximadamente: calcula el ancho preferido para el formato del contenido sin los saltos de línea con excepción de donde hay saltos de líneas explícitas, y también calcula el ancho mínimo preferido, es decir, intentando todos los saltos de líneas posibles. CSS 2.1 no define el algoritmo exacto. En tercer lugar, encuentra *el ancho disponible*: en este caso, éste es el ancho del bloque de contención menos los valores usados en 'margin-left', 'border-left-width', 'padding-left', 'padding-right', 'border-right-width', 'margin-right' y el ancho de cualquier barra de desplazamiento relevante.

Entonces encoger al intervalo del ancho es: $\min(\max(\text{ancho mínimo preferido}, \text{ancho disponible}), \text{ancho preferido})$.

10.3.6 Elementos reemplazados, Flotantes

Si 'margin-left' o 'margin-right' son computados como 'auto', sus valores usados son '0'. El valor usado de 'width' es determinado por los [elemento reemplazados en línea](#).

10.3.7 Elementos no reemplazados, posicionamiento absoluto

Para lo propuesto en esta sección y siguientes, el termino “posición estática” (de un elemento) se refiere, aproximadamente, a la posición que un elemento tendría en el flujo normal. Más preciso:

- La posición estática (static position) para 'left' es la distancia desde el borde izquierdo del bloque de contención hasta el borde del margen izquierdo de una caja hipotética que tendría que ser la primera caja del elemento si su propiedad 'position' fuese 'static'. El valor es negativo si la caja hipotética está a la izquierda del bloque de contención.
- La posición estática para 'right' es la distancia desde el borde derecho del bloque de contención hasta el borde del margen derecho de una caja hipotética como en el caso anterior. El valor es positivo si la caja hipotética está a la izquierda del borde del bloque de contención.

Pero realmente para el calculo de las dimensiones de esa caja hipotética, las aplicaciones de usuarios son libres de hacer conjeturas para la posición más probable

Para el objetivo de calcular la posición estática, el bloque de contención de elementos posicionados fijos es el bloque de contención inicial en lugar del acceso visual.

La condición que determina el valor usado para estos elementos es:

`'left' + 'margin-left' + 'border-left-width' + 'padding-left' + 'width' + 'padding-right' + 'border-right-width' + 'margin-right' + 'right' = ancho del bloque de contención`

Si 'left', 'width', y 'right' son 'auto': Primero fija para cualquier valor 'auto' de 'margin-left' y 'margin-right' a cero. Luego, si 'direction' es 'ltr' fija 'left' a [posición estática](#) y aplica la regla tres de abajo, sino, fija 'right' a [posición estática](#) y aplica la regla numero uno de abajo.

Si ninguna de las tres es 'auto': Si tanto 'margin-left' como 'margin-right' son 'auto', se resuelve la ecuación bajo la condición de que los dos márgenes toman el mismo valor. Si uno de 'margin-left' o 'margin-right' es 'auto', se soluciona la ecuación para ese valor. Si los valores son sobre-obligados, se ignora el valor para 'left' (en caso de 'direction' con valor 'rtl') o 'right' (en caso de 'direction' con valor 'ltr') y se resuelve la ecuación para ese valor.

Si no, se fija el valor 'auto' para 'margin-left' y 'margin-right' a 0, y se elige una de las siguientes reglas que se aplican:

1. 'left' y 'width' son 'auto' y 'right' es distinto de 'auto', entonces el ancho es reducido al intervalo. después se resuelve 'left'
2. 'left' y 'right' son 'auto' y 'width' es distinto de 'auto', entonces si 'direction' es 'ltr' establece 'left' al [posicionamiento estático](#), si no se establece 'right' al [posicionamiento estático](#). Luego se soluciona para 'left' (si 'direction' es 'rtl') o 'right' (si 'direction' es 'ltr').
3. 'width' y 'right' son 'auto' y 'left' es distinto de 'auto', entonces el ancho es reducido al intervalo. Luego se resuelve para 'right'
4. 'left' es 'auto', 'width' y 'right' son distintos de 'auto', entonces se resuelve para 'left'
5. 'width' es 'auto', 'left' y 'right' son distintos de 'auto', entonces se resuelve para 'width'
6. 'right' es 'auto', 'left' y 'width' son distintos de 'auto', entonces se resuelve para 'right'

El “encogimiento al intervalo” del ancho se calcula igual que el ancho de una celda de una tabla utilizando el algoritmo automático de representación de la tabla. Aproximadamente: calcula el ancho preferido para el formato del contenido sin los saltos de línea con excepción de donde hay saltos de líneas explícitas, y también calcula el ancho mínimo preferido, es decir, intentando todos los saltos de líneas posibles. CSS 2.1 no define el algoritmo exacto. En tercer lugar, encuentra *el ancho disponible* : este se encuentra resolviendo ‘width’ después de establecer ‘left’ (en el caso 1) o ‘right’ (en el caso 3) a 0.

Entonces el encogimiento al intervalo del ancho es: $\min(\max(\text{ancho mínimo preferido}, \text{ancho disponible}), \text{ancho preferido})$.

10.3.8 Elementos reemplazados, posicionados absolutamente

Esta situación es similar a la previa, excepto que el elemento tiene un ancho [intrínseco](#). La secuencia de sustituciones es ahora:

1. El valor usado de ‘width’ es determinado como para [elementos reemplazados en línea](#).
2. Si ‘left’ tiene el valor ‘auto’ mientras que ‘direction’ es ‘ltr’, se reemplaza ‘auto’ con la [posición estática](#).
3. Si ‘right’ tiene el valor ‘auto’ mientras que ‘direction’ es ‘rtl’, se reemplaza ‘auto’ con la [posición estática](#).
4. Si ‘left’ o ‘right’ son ‘auto’, se reemplaza cualquier ‘auto’ para ‘margin-left’ o ‘margin-right’ con ‘0’.
5. Si en este punto ambos, ‘margin-left’ y ‘margin-right’ son también ‘auto’, se resuelve la ecuación bajo la exigencia de que dos márgenes deben tener iguales valores.
6. Si en este punto queda un sólo ‘auto’, se resuelve la ecuación para ese valor.
7. Si en este punto el valor está sobre-obligado, se ignora el valor para ‘left’ (en caso de ‘direction’ sea ‘rtl’) o ‘right’ (en caso de que ‘direction’ sea ‘ltr’) y se resuelve ese valor.

10.3.9 Elementos no reemplazados en flujo normal, 'bloques en línea'

Si ‘width’ es ‘auto’, el valor usado es el de [encoger al intervalo](#) el ancho como para elementos flotantes.

Un valor computado de ‘auto’ para ‘margin-left’ o ‘margin-right’ se convierte en un valor usado de ‘0’.

10.3.10 Elementos reemplazados en flujo normal, 'Bloques en línea',

Igual que para los [elementos reemplazados en línea](#)

10.4 Anchos mínimos y máximos: 'min-width' y 'max-width'

'min-width'

<i>Valor:</i>	<u><medida></u> <u><porcentaje></u> <u>inherit</u>
<i>Inicial:</i>	0
<i>Se aplica a:</i>	Todos los elementos excepto elementos no reemplazados en línea y elementos tabla
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	Referidos al ancho del bloque de contención
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	El porcentaje especificado o la medida absoluta

'max-width'

<i>Valor:</i>	<u><medida></u> <u><porcentaje></u> none <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Todos los elementos excepto elementos no reemplazados en línea y elementos tabla
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	Referidos al ancho del bloque de contención
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	El porcentaje especificado o la medida absoluta o 'none'

Estas dos propiedades permiten a los autores restringir los anchos de las cajas a cierto rango. Los valores tienen los siguientes significados:

<medida>

Especifica un ancho mínimo o máximo fijo usado.

<porcentaje>

Especifica un porcentaje para determinar el valor usado. El porcentaje es calculado con respecto al ancho del [bloque de contención](#) de la caja generada.

none

(solo en 'max-width') Ninguna limitación en el ancho de la caja.

Valores negativos para 'min-width' y 'max-width' son ilícitos.

El siguiente algoritmo describe como influyen las dos propiedades en el [valor usado](#) de la propiedad 'width':

1. El ancho provisional usado es calculado (sin 'min-width' y 'max-width') siguiendo las reglas descritas en "[Computando anchos y márgenes](#)" más arriba.
2. Si el ancho provisional usado es mayor que 'max-width', las reglas de [arriba](#) son aplicadas otra vez, pero esta vez usando el valor computado de 'max-width' como el valor computado para 'width'.
3. Si el ancho resultante es mas pequeño que 'min-width', las reglas de [arriba](#) son aplicadas otra vez, pero esta vez usando el valor de 'min-width' como el valor computado para 'width'.

Sin embargo, para elementos reemplazados con ambos 'width' y 'height' especificados como 'auto', el algoritmo es el siguiente:

Seleccionamos de la tabla los valores resueltos del alto y ancho para violación de la restricción apropiada. Toma el max-width y max-height como max(min, max) para que $\min \leq \max$ sea verdadero. En esta tabla, w y h significan la anchura y altura intrínsecas, respectivamente.

Violación de la Obligación	Ancho Resuelto	Alto Resuelto
none	w	H
$w > \text{max-width}$	max-width	$\max(\text{max-width} * h/w, \text{min-height})$
$w < \text{min-width}$	min-width	$\min(\text{min-width} * h/w, \text{max-height})$
$h > \text{max-height}$	$\max(\text{max-height} * w/h, \text{min-width})$	max-height
$h < \text{min-height}$	$\min(\text{min-height} * w/h, \text{max-width})$	Min-height
$(w > \text{max-width})$ y $(h > \text{max-height})$, donde $(\text{max-width}/w \leq \text{max-height}/h)$	max-width	$\max(\text{min-height}, \text{max-width} * h/w)$
$(w > \text{max-width})$ y $(h > \text{max-height})$, donde $(\text{max-width}/w > \text{max-height}/h)$	$\max(\text{min-width}, \text{max-height} * w/h)$	max-height
$(w < \text{min-width})$ y $(h < \text{min-height})$, donde $(\text{min-width}/w \leq \text{min-height}/h)$	$\min(\text{max-width}, \text{min-height} * w/h)$	Min-height
$(w < \text{min-width})$ y $(h < \text{min-height})$, donde $(\text{min-width}/w > \text{min-height}/h)$	min-width	$\min(\text{max-height}, \text{min-width} * h/w)$
$(w < \text{min-width})$ y $(h > \text{max-height})$	min-width	max-height
$(w > \text{max-width})$ y $(h < \text{min-height})$	max-width	Min-height

Luego se aplican las reglas de ["Computando anchos y márgenes"](#) de arriba, si 'width' fuera computado como este valor.

10.5 Altura del contenido: la propiedad 'height'

'height'

Valor: <medida> | <porcentaje> | auto | inherit

Inicial: auto

Se aplica a: Todos los elementos excepto a elementos no reemplazados en línea, columnas de tabla y grupos de columna

Se hereda: no

Porcentajes: Ver el texto

Medio: visual

Valor computado: El porcentaje o 'auto' como el especificado o la medida absoluta; 'auto' si la propiedad no se aplica

Esta propiedad especifica la [altura del contenido](#) de las cajas generadas por los elementos a nivel de bloque, bloques en línea y [reemplazados](#).

Esta propiedad no se aplica a los elementos no reemplazados [a nivel de línea](#). Ver la [sección sobre computando alturas y márgenes para elementos no reemplazados en línea](#) para las reglas usadas.

Los valores tienen los siguientes significados:

<medida>

Especifica la altura del área de contenido usando un valor de medida.

<porcentaje>

Especifica una altura en porcentaje. El porcentaje es calculado con respecto a la altura del [bloque de contención](#) de la caja generada. Si la altura del bloque de contención no es especificada explícitamente (es decir, depende de la altura del contenido), y este elemento no está posicionado absolutamente, el valor es interpretado como 'auto'. Una altura en porcentaje de un [elemento raíz](#) es relativa al [acceso visual](#).

auto

La altura depende de los valores de otras propiedades. Ver el texto más abajo.

Observe que la altura del bloque de contención de un elemento posicionado absolutamente depende del tamaño de dicho elemento, y así una altura en porcentaje sobre dicho elemento puede ser siempre resuelta. Sin embargo, puede que esa altura no se conozca antes de los elementos que se colocan más tarde en el documento que es procesado.

Los valores negativos para `'height'` son ilícitos.

Por ejemplo, la siguiente regla establece la altura del contenido del párrafo a 100 píxeles:

```
p { height: 100px }
```

Los párrafos que requieren más de 100px de altura se desbordarán de acuerdo a la propiedad `'overflow'`.

10.6 Computando alturas y márgenes

Para calcular los valores de `'top'`, `'margin-top'`, `'height'`, `'margin-bottom'`, y `'bottom'` debe hacerse una distinción entre los distintos tipos de cajas:

1. Elementos no reemplazados, en línea
2. Elementos reemplazados, en línea
3. Elementos no reemplazados en flujo normal, a nivel de bloque
4. Elementos reemplazados en flujo normal, a nivel de bloque
5. Elementos no reemplazados, flotantes
6. Elementos reemplazados, flotantes
7. Elementos no reemplazados, posicionados absolutamente
8. Elementos reemplazados, posicionados absolutamente
9. Elementos no reemplazados en flujo normal, 'Bloques en línea'
10. Elementos reemplazados en flujo normal, 'Bloques en línea'

Para los puntos 1-6 y 9-10, el valor usado de `'top'` y `'bottom'` se determinan por las reglas en la sección 9.4.3.

10.6.1 En línea, elementos no reemplazados

La propiedad 'height' no se aplica. El alto del área de contenido estará basado en la fuente, pero esta especificación no indica como. Una AU puede, por ejemplo utilizar las fuentes de las cajas em o el máximo del antepasado o del descendente. (Esto podría asegurar que los glyphs con partes encima o debajo de em-boxwith caen dentro del área de contenido, pero conduce a tamaños diferentes de cajas para diferentes fuentes, lo anterior asegura que los autores puedan controlar el estilo del fondo en relación a 'line-height', pero conduce a glyphs pintados fuera de su área de contenido).

Nota: El nivel 3 de CSS probablemente incluirá una propiedad para seleccionar que tipo de fuente es utilizada para el alto del contenido.

Nota del traductor: Las formas que una fuente usa para representar los caracteres de una cadena son llamados **glyphs**

El relleno vertical, borde y margen de una caja no reemplazada en línea empieza en la parte superior e inferior del área de contenido, no el 'line-height' (alto de línea). Pero solo el 'line-height' es utilizado para calcular el alto de la línea de la caja.

Si se utiliza mas de una fuente (esto puede ocurrir cuando los glyphs se encuentran en diferentes fuentes), el alto del área de contenido no está definido en esta especificación. Sin embargo, sugerimos que la altura sea elegida tal que el área de contenido sea lo bastante alto para (1) las cajas em, o (2) el máximo de los antepasados y descendentes, de todas las fuentes del elemento. Observe que esto puede ser más grande que cada uno de los tamaños de las fuentes involucradas, dependiendo de la alineación de la línea base de las fuentes.

10.6.2 Elementos reemplazados en línea, elementos reemplazados a nivel de bloque en flujo normal, elementos reemplazados de 'bloques en línea' en flujo normal y elementos reemplazados flotantes

Si 'margin-top', o 'margin-bottom' son 'auto', sus valores usados son 0. Si 'height' tiene un valor computado de 'auto' y 'width' también tiene un valor computado de 'auto', la altura intrínseca del elemento es la utilizada para 'height'. Si 'height' tiene un valor computado de 'auto' y 'width' tiene otro valor computado, entonces el valor usado para 'height' es:

$$(\text{altura intrínseca}) * ((\text{ancho usado}) / (\text{ancho intrínseco}))$$

10.6.3 Elementos a nivel de bloque no reemplazados en flujo normal cuando 'overflow' es computado a 'visible'

Si 'margin-top', o 'margin-bottom' son 'auto', sus valores usados son 0. Si 'height' es 'auto', la altura depende de si el elemento tiene algún hijo a nivel de bloque y si tiene relleno o bordes:

Si solo tiene un hijo a nivel de línea, la altura es la distancia entre la parte superior de la línea mas alta de la caja y la parte inferior de la línea más baja de la caja.

Si tiene hijos a nivel de bloque, la altura es la distancia entre el borde superior de la parte más alta de la caja hija a nivel de bloque que no tiene [márgenes cerrados](#) y el borde inferior de la parte más baja de la caja hija a nivel de bloque que no tiene márgenes cerrados. Sin embargo, si los elementos tienen un relleno y/o borde superior distinto de cero, o es el elemento raíz, entonces el contenido empieza en el borde del margen superior del hijo situado más arriba. (El primer caso expone el hecho de que los márgenes superior e inferior del elemento se [cierran](#) con los más superiores e inferiores del hijo, mientras en el segundo caso la presencia de relleno/borde previene el margen superior de los [cerrados](#).) Igual, si el elemento tiene relleno inferior y/o borde distinto de cero, entonces el contenido termina en el borde del margen inferior del hijo situado más abajo.

Solo los hijos en el flujo normal son tomados en cuenta (es decir, cajas flotantes y cajas posicionadas absolutamente son ignoradas, y cajas posicionadas relativamente son consideradas sin su posición). Observe que la caja hija puede ser una [caja de bloque anónima](#).

10.6.4 Elementos no reemplazados, posicionamiento absoluto

Para lo propuesto en esta sección y la siguiente, el término “posición estática” (static position) (de un elemento) se refiere, aproximadamente, a la posición que un elemento tendría en el flujo normal. Más precisamente, la posición estática para 'top' es la distancia desde el borde superior del bloque de contención hasta el borde del margen superior de una caja hipotética que sería la primera caja del elemento como si su propiedad 'position' fuese 'static'. El valor es negativo si la caja hipotética está encima del bloque de contención.

Pero en realidad para el cálculo de las dimensiones de esa caja hipotética, las aplicaciones de usuarios son libres para hacer las conjeturas sobre esa posición probable

Para los objetivos de calcular la posición estática, el bloque de contenido de los elementos posicionados como fijos es el bloque de contenido inicial colocado en el acceso visual

Para elementos posicionados absolutamente, el valor usado para las dimensiones verticales deben satisfacer esta condición:

$'top' + 'margin-top' + 'border-top-width' + 'padding-top' + 'height' + 'padding-bottom' + 'border-bottom-width' + 'margin-bottom' + 'bottom' = \text{alto del bloque de contenido}$

Si los tres 'top', 'height', y 'bottom' son auto, se pone 'top' como posición estática y se aplica la regla número tres de abajo.

Si ninguna de las tres son 'auto': Si tanto 'margin-top' y 'margin-bottom' son 'auto', se resuelve la ecuación bajo la condición extra de dos márgenes con valores iguales. Si 'margin-top' o 'margin-bottom' es 'auto', se resuelve la ecuación para ese valor. Si los valores son sobre obligados, se ignora el valor para 'bottom' y se resuelve para ese valor.

Si no, se aplica una de las siguientes seis reglas:

1. 'top' y 'height' son 'auto' y 'bottom' es distinto de 'auto', entonces el ancho es [basado sobre el contenido](#), se fijan los valores 'auto' para 'margin-top' y 'margin-bottom' a 0, y se resuelve para 'top'

2. 'top' y 'bottom' son 'auto' y 'height' es distinto de 'auto', entonces se fija 'top' a posición estática, se fijan los valores 'auto' para 'margin-top' y 'margin-bottom' a 0, y se resuelve para 'bottom'
3. 'height' y 'bottom' son 'auto' y 'top' distinto de 'auto', entonces la altura es [basada sobre el contenido](#), se fijan los valores 'auto' para 'margin-top' y 'margin-bottom' a 0, y se resuelve para 'bottom'
4. 'top' es 'auto', 'height' y 'bottom' son distintos de 'auto', entonces se fijan los valores 'auto' para 'margin-top' y 'margin-bottom' a 0, y se resuelve para 'top'
5. 'height' es 'auto', 'top' y 'bottom' son distintos de 'auto', entonces los valores autos para 'margin-top' y 'margin-bottom' son fijados a 0 y se resuelve para 'height'
6. 'bottom' es 'auto', 'top' y 'height' son distintos de 'auto', entonces se fijan los valores 'auto' para 'margin-top' y 'margin-bottom' a 0 y se resuelve para 'bottom'

10.6.5 Elementos reemplazados, posicionados absolutamente

Esta situación es similar a la vista anteriormente, excepto que los elementos tienen una altura [intrínseca](#). La secuencia de sustituciones es ahora:

1. El valor utilizado de 'height' es determinado igual que para [elementos reemplazados en línea](#).
2. Si 'top' tiene el valor 'auto', se reemplaza con la posición estática (static) del elemento.
3. Si 'bottom' es 'auto', se reemplaza cualquier 'auto' en 'margin-top' o 'margin-bottom' con '0'.
4. Si en este punto tanto 'margin-top' como 'margin-bottom' son también 'auto', se resuelve la ecuación bajo la condición adicional de que los dos márgenes deben tener igual valor.
5. Si en este punto hay solo un 'auto', se resuelve la ecuación para ese valor.
6. Si en este punto los valores son sobre-obligados, se ignora el valor de 'bottom' y se resuelve la ecuación para ese valor.

10.6.6 Nivel de bloque, elementos no reemplazados en flujo normal cuando 'overflow' no es computado como 'visible'; 'bloques en línea', elementos no reemplazados; y flotantes, elementos no reemplazados

Si 'margin-top', o 'margin-bottom' son 'auto', su valor utilizado es 0. Si 'height' es 'auto', la [altura depende de los descendentes del elemento](#).

10.6.7 Alturas 'Auto' para el contexto de formato en bloque raíz

En ciertos casos (ver la sección precedente), la altura de un elemento puede computarse como sigue:

Si solo tiene hijos a nivel de línea, la altura es la distancia entre la parte superior mas alta de la línea de la caja y la parte inferior mas baja de la línea de la caja.

Si tiene hijos a nivel de bloque, la altura es la distancia entre el borde del margen superior de la parte mas alta de la caja hija a nivel de bloque y el borde del margen inferior de la parte mas baja de la caja hija a nivel de bloque.

Los hijos posicionados absolutamente son ignorados, y las cajas posicionadas relativamente son consideradas sin su posición. Observe que la caja hija puede ser una [caja de bloque anónima](#).

Además, si el elemento tiene algún descendente flotante cuyo borde de margen inferior está debajo de la parte inferior, entonces la altura es incrementada para incluir esos bordes. Solo las flotantes que sean hijas de dicho elemento o de descendentes en el flujo normal son tomadas en cuenta. Por ejemplo, flotantes dentro de descendentes posicionadas absolutamente u otras flotantes que no lo están.

10.7 Alturas mínimas y máximas: 'min-height' y 'max-height'

A veces resulta útil restringir la altura de los elementos a cierto rango. Dos propiedades ofrecen esta función:

'min-height'

<i>Valor:</i>	<u><medida></u> <u><porcentaje></u> <u>inherit</u>
<i>Inicial:</i>	0
<i>Se aplica a:</i>	Todos los elementos excepto los elementos en línea no reemplazados y elementos tabla
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	Ver texto
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	El porcentaje especificado o la medida absoluta

'max-height'

<i>Valor:</i>	<u><medida></u> <u><porcentaje></u> <u>none</u> <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Todos los elementos excepto los elementos en línea no reemplazados y elementos tabla
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	Ver texto
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	El porcentaje especificado o la medida absoluta o 'none'

Estas dos propiedades permite a los autores restringir la altura de las cajas a un cierto rango. Los valores tiene los siguientes significados:

<medida>

Especifica una altura mínima o máxima computada fija

<porcentaje>

Especifica un porcentaje para determinar el valor usado. El porcentaje es calculado con respecto a la altura del [bloque de contención](#) de la caja generada. Si la altura del bloque de contención no está especificada explícitamente (es decir, depende de la altura del contenido), el valor del porcentaje es interpretado como '0' (para 'min-height') o 'none' (para 'max-height').

none

(Solo en 'max-height') Ninguna limitación en la altura de la caja

Valores negativos para 'min-height' y 'max-height' son ilícitos.

El siguiente algoritmo describe como las dos propiedades influyen en el [valor computado](#) de la propiedad `'height'`:

1. La altura provisional utilizada es calculada (sin `'min-height'` y `'max-height'`) siguiendo las reglas sobre ["Computando altos y márgenes"](#) de más arriba.
2. Si la altura provisional es mas grande que `'max-height'`, las reglas [anteriores](#) son aplicadas nuevamente, pero esta vez usando el valor de `'max-height'` como el valor computado para `'height'`.
3. Si la altura resultante es mas pequeña que `'min-height'`, las reglas [anteriores](#) son aplicadas nuevamente, pero esta vez usando el valor de `'min-height'` como el valor computado para `'height'`.

Sin embargo, para elementos reemplazados con `'width'` y `'height'` computados como 'auto', utilizan el algoritmo sobre [anchos mínimos y máximos](#) de arriba para encontrar el ancho y alto utilizado. Luego aplica las reglas sobre ["Computando alturas y márgenes"](#) de arriba, usando la resultante anchura y altura como si fueran los valores computados.

10.8 Calculando la altura de la línea: propiedades `'line-height'` y `'vertical-align'`

Como se describe en la sección sobre [contexto de formatos en línea](#), en las aplicaciones del usuario fluyen las cajas en línea dentro de una pila (apilamiento) vertical de [línea de cajas](#). La altura de una línea de caja es determinada como sigue:

1. La altura de cada caja en línea en la línea de la caja es calculada (ver ["Computando anchos y márgenes"](#) y la propiedad `'line-height'`).
2. Las cajas en línea son alineadas verticalmente de acuerdo a su propiedad `'vertical-align'`.
3. La altura de la línea de caja es la distancia entre la caja ubicada mas arriba (en la pila) y la caja ubicada mas abajo (en la pila).

Los elementos en línea vacíos generan cajas en línea vacías, pero estas cajas aun tienen márgenes, relleno, bordes y altura de línea, y así influyen en estos cálculos tanto como los elementos con contenido.

10.8.1 Interlineado y medio interlineado

Ya que el valor de `'line-height'` puede ser diferente de la altura del área de contenido puede haber espacio encima y debajo de los glyphs procesados. La diferencia entre la altura del contenido y el valor usado de `'line-height'` es llamado *interlineado*. La mitad del interlineado es llamado medio interlineado.

Las aplicaciones de usuarios centran verticalmente los glyphs en una caja en línea, añadiendo medio interlineado en la parte superior e inferior. Por ejemplo, si un fragmento de texto es de '12px' de alto y el valor de `'line-height'` es '14px', 2pxs de espacio extra deberían ser añadidos: 1px encima y 1px debajo de las letras. (Esto se aplica también a las cajas vacías, como si la caja vacía contuviera una letra infinitamente condensada.)

Cuando el valor de '[line-height](#)' es menor que la altura del contenido, la altura final de la caja en línea será menor que el tamaño de la fuente y los glyphs procesados serán “sangrados” fuera de la caja. Si una de esas cajas tocan el borde de la línea de la caja, los glyphs serán “sangrados” dentro de la línea de la caja adyacente.

Aunque los márgenes, bordes y rellenos de los elementos no reemplazados no entran en el cálculo de la línea de la caja, son también procesados alrededor de las cajas en línea. Esto significa que si la altura especificada por '[line-height](#)' es menor que la altura del contenido de las cajas que contiene, los fondos y colores del relleno y borde pueden “sangrarse” dentro de las líneas de las cajas. Las aplicaciones del usuario pueden procesar las cajas en orden al documento. Este será el caso de bordes sobre líneas de secuencias pintadas sobre los bordes y texto de líneas anteriores.

'line-height'

<i>Valor:</i>	normal <número> <medida> <porcentaje> inherit
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	Referidos al tamaño de la fuente del propio elemento
<i>Medio:</i>	visual
<i>Valor computado:</i>	Para <medida> y <porcentaje> el valor absoluto; sino el especificado

Si la propiedad es puesta en un elemento a [nivel de bloque](#), table-cell, table-caption o bloque en línea cuyo contenido esté compuesto de elementos a [nivel de línea](#), se especifica la altura mínima de cajas de línea dentro del elemento. La altura mínima consiste en la altura mínima encima de la línea base del bloque y una profundidad mínima debajo de ella, exactamente como si cada línea comenzará con una anchura cero de la caja en línea con la fuente del bloque y las propiedades de la altura de línea (Este T_EX se llama "strut").

Si la propiedad es puesta en un elemento a [nivel de línea](#), se especifica la altura que es usada en el calculo de la altura de la caja de line (excepto para los elementos en línea [reemplazados](#), donde la altura de la caja es dada por la propiedad '[height](#)').

Los valores para esta propiedad tiene los siguientes significados:

normal

Le dice a las aplicaciones del usuario que ponga como valor usado un valor “razonable” en base a la fuente del elemento. El valor tiene el mismo significado que [<número>](#). Recomendamos un valor usado para 'normal' entre 1.0 y 1.2. El [valor computado](#) es 'normal'.

<medida>

La medida especificada es usada en el calculo de la altura de la caja de línea. Los valores negativos son ilícitos.

<número>

El valor usado de la propiedad es este número multiplicado por el tamaño de la fuente del elemento. Valores negativos son ilícitos. El [valor computado](#) es el mismo que el valor especificado.

<porcentaje>

El [valor computado](#) de la propiedad es este porcentaje multiplicado por el tamaño computado de la fuente del elemento. Los valores negativos son ilícitos.

Las tres reglas en el ejemplo de abajo tienen como resultado la misma altura de línea:

```
div { line-height: 1.2; font-size: 10pt } /* número */
div { line-height: 1.2em; font-size: 10pt } /* medida */
div { line-height: 120%; font-size: 10pt } /* porcentaje */
```

Cuando un elemento contiene texto que es procesado como más de una fuente, las aplicaciones del usuario deben determinar el valor de '[line-height](#)' de acuerdo al tamaño de la fuente mayor.

Generalmente, cuando hay solo un valor de '[line-height](#)' para todas las cajas en línea en el párrafo (y ninguna imagen alta), lo anterior asegurará que las líneas base de sucesivas líneas están exactamente '[line-height](#)' separadas. Esto es importante cuando las columnas de texto en diferentes fuentes tienen que ser alineadas, por ejemplo en una tabla.

'vertical-align'

Valor: baseline | sub | super | top | text-top | middle | bottom | text-bottom | [<porcentaje>](#) | [<medida>](#) | [inherit](#)

Inicial: baseline

Se aplica a: Elementos a nivel de línea y 'table-cell'

Se hereda: no

Porcentaje: Referidos al 'line-height' del propio elemento

Medio: [visual](#)

Valor computado: Para [<porcentaje>](#) y [<medida>](#) la medida absoluta, si no como el especificado

Esta propiedad afecta al posicionamiento vertical dentro de una caja de línea de las cajas generadas por un elemento a nivel de línea.

Nota. Los valores de esta propiedad tienen significados ligeramente distintos en el contexto de tablas. Por favor consulte la sección sobre [algoritmos para la altura de tabla](#) para más detalles.

Los siguientes valores solo tienen significado con respecto a un elemento padre a nivel de línea, o al [punta](#) de un elemento padre a [nivel de bloque](#), table-cell, table-caption o bloque en línea:

baseline

Alinea la línea base de la caja con la línea base de la caja padre. Si la caja no tiene una línea base, alinea el borde del margen inferior con la línea base del padre.

middle

Alinea el punto medio vertical de la caja con la línea base de la caja padre más la mitad de la altura x del padre.

sub

Baja la línea base de la caja hasta la posición apropiada para subíndices en la caja padre. (Este valor no tiene efecto sobre el tamaño de la fuente del texto del elemento).

super

Eleva la línea base de la caja hasta la posición adecuada para superíndices de la caja padre. (Este valor no tiene efecto sobre el tamaño de la fuente del texto del elemento).

text-top

Alinea la parte superior de la caja con la parte superior de la fuente del elemento padre

text-bottom

Alinea la parte inferior de la caja con la parte inferior de la fuente del elemento padre.

<porcentaje>

Eleva (valor positivo) o baja (valor negativo) la caja en esta distancia (un porcentaje del valor 'line-height'). El valor '0%' significa lo mismo que 'baseline'.

<medida>

Eleva (valor positivo) o baja (valor negativo) la caja en esta distancia. El valor '0cm' significa lo mismo que 'baseline'.

Los siguientes valor alinean el elemento en relación a la línea de la caja. Ya que el elemento puede tener hijos alineados en relación a el (que en su vez puede tener descendentes alineado en relación a ellos), estos valores utilizan los límites de la subestructura alineada. La subestructura alineada de un elemento en línea contiene ese elemento y las subestructuras alineados de todos los elementos hijos en línea cuyo valor computado 'vertical-align' no es 'top' o 'bottom'. La parte superior de la subestructura alineada es la más alta de la parte alta de las cajas en la subestructura, y la parte inferior es análoga.

top

Alinea la parte superior de la subestructura con la parte superior de la caja de línea.

bottom

Alinea la parte inferior de la subestructura con la parte inferior de la caja de línea.

La línea base de una 'inline-table' (tabla en línea) es la línea base de la primera fila de la tabla.

La AU puede utilizar la línea base de la última línea de la caja en el flujo normal en el elemento como la línea base de un 'bloque en línea', o el borde del margen inferior del elemento, si no hay.

11 Efectos visuales

Contenido

- [11.1 Desbordamiento y recorte](#)
 - [11.1.1 Desbordamiento: la propiedad 'overflow'](#)
 - [11.1.2 Recorte: la propiedad 'clip'](#)
- [11.2 Visibilidad: la propiedad 'visibility'](#)

11.1 Desbordamiento y recorte

Generalmente, el contenido de una caja de bloque se confina a los límites del contenido de la caja. En ciertos casos, una caja puede *desbordar*, significando que su contenido queda parcial o completamente fuera de la caja, por ejemplo:

- Una línea no puede ser cortada, provocando que la caja de línea sea más ancha que la caja de bloque.
- Una caja a nivel de bloque es demasiado ancha para el bloque de contención. Esto puede suceder cuando la propiedad `'width'` de un elemento tiene un valor que provoca que la caja de bloque generada se salga por los lados del bloque de contención.
- La altura de un elemento excede la altura explícitamente asignada al bloque de contención (es decir, la altura del bloque de contención es determinada por la propiedad `'height'`, no por la altura del contenido).
- Una caja descendiente es [posicionada absolutamente](#), parcialmente fuera de la caja. Tales cajas no siempre son cortadas por la propiedad de desbordamiento de sus ascendentes.
- Una caja descendiente tiene [márgenes negativos](#), provocando que sea posicionada parcialmente fuera de la caja.
- La propiedad `'text-indent'` provoca que una caja en línea sea colgada del límite izquierdo o derecho de la caja de bloque.

Siempre que ocurra el desbordamiento, la propiedad `'overflow'` especifica si una caja es recortada a la caja que la contiene, y si es así, si se proporciona un mecanismo de desplazamiento para acceder al contenido recortado.

11.1.1 Desbordamiento: la propiedad `'overflow'`

'overflow'

<i>Valor:</i>	visible hidden scroll auto <u>inherit</u>
<i>Inicial:</i>	visible
<i>Se aplica a:</i>	Nivel de bloque y elementos reemplazados, celdas de tabla, bloques en línea
<i>Se Hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medios:</i>	<u>visual</u>
<i>Valor computado:</i>	según el especificado

Esta propiedad especifica si el contenido de un elemento a nivel de bloque es recortado cuando desborda la caja del elemento. Esto afecta al recorte de todo el contenido del elemento excepto

algunos elementos descendentes (y su respectivo contenido y descendentes) cuyo bloque de contención es el acceso visual o un ascendente del elemento. Los valores tienen los siguientes significados:

visible

Este valor indica que el contenido no está recortado, es decir, puede ser procesado fuera de la caja de bloque.

hidden

Este valor indica que el contenido es recortado y que ningún interfaz del usuario debe proporcionar barras de desplazamiento para ver el contenido fuera de la región recortada.

scroll

Este valor indica que el contenido es recortado y que si la aplicación del usuario utiliza un mecanismo de desplazamiento que sea visible en la pantalla (tal como una barra de desplazamiento o un panner), ese mecanismo se debe mostrar en una caja tenga o no ésta parte de su contenido recortado. Esto evita cualquier problema con las barras de desplazamiento que aparecen y que desaparecen en un entorno dinámico. Cuando se especifica este valor y el medio al que está dirigido es 'print', el contenido que se desborda puede ser impreso.

auto

El comportamiento del valor 'auto' depende de la aplicación del usuario, pero es usuario agente-dependiente, pero debe provocar que se proporcione un mecanismo de desplazamiento para las cajas desbordadas.

Incluso si 'overflow' se fija como 'visible', el contenido puede ser recortado a la ventana del documento de una AU por el entorno de funcionamiento nativo.

Las AUs de HTML pueden aplicar la propiedad del desbordamiento de los elementos del BODY o del HTML en el acceso visual.

En el caso de una barra de desplazamiento colocada en un límite de la caja del elemento, ésta debe ser insertada entre el límite del borde interno y el borde externo del relleno. Cualquier espacio tomado por las barras de desplazamiento se debe restar del width/height computado, así se preserva el límite del borde interno.

Considere el ejemplo siguiente de un bloque de cita (<blockquote>) que es demasiado grande para su bloque de contención (establecido por un <div>) Aquí está el documento fuente:

```
<div>
<blockquote>
<p>No me gustó la obra, pero la ví
bajo condiciones adversas -el telón estaba levantado-.</p >
<cite>- Groucho Marx</cite>
</blockquote>
</div >
```

Aquí está la hoja de estilo que controla los tamaños y el estilo de las cajas generadas:

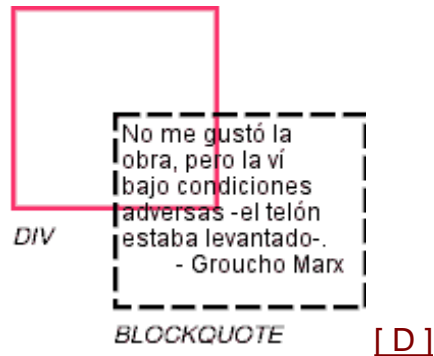
```
div { width : 100px; height: 100px;
      border: thin solid red;
      }
blockquote { width : 125px; height : 100px;
```

```

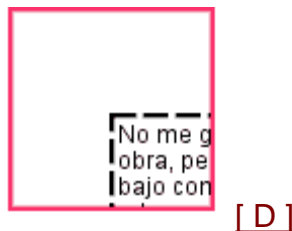
margin-top: 50px; margin-left: 50px;
border: thin dashed black
}
cite { display: block;
text-align : right;
border: none
}

```

El valor inicial de 'overflow' es 'visible', tal que `<blockquote>` sea procesado sin recortar, algo como esto:



Fijando 'overflow' como 'hidden' para `<div>`, por otra parte, provoca que `<blockquote>` sea recortado por el bloque de contención:



Un valor de 'scroll' le indicaría a las AUs que soporten mecanismos de desplazamiento visibles que muestren uno para que los usuarios puedan tener acceso al contenido recortado.

Finalmente, considere este caso donde un elemento posicionado absolutamente se mezcla con un padre de desbordado.

Hoja de estilo:

```

container { position: relative; border: solid; }
scroller { overflow: scroll; height: 5em; margin: 5em; }
satellite { position: absolute; top: 0; }
body { height: 10em; }

```

Fragmento del documento:

```

<container>
  <scroller>
    <satellite/>
  </scroller>
</container>

```


En este ejemplo, el elemento "scroller" no desplaza al elemento "satellite", porque el último bloque de contención está fuera del elemento cuyo desbordamiento está siendo recortado y desplazado.

11.1.2 Recorte: la propiedad **'clip'**

Una zona de recorte define qué porción del borde de la caja del elemento es visible. Por defecto, la región de recorte tiene el mismo tamaño y la forma que el borde de la caja del elemento. Sin embargo, la región de recorte se puede modificar por la propiedad **'clip'**.

'clip'

<i>Valor:</i>	<u><forma></u> <u>auto</u> <u>inherit</u>
<i>Inicial:</i>	Auto
<i>Se aplica a:</i>	elementos posicionados absolutamente
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medios:</i>	<u>visual</u>
<i>Valor computado:</i>	Para los valores del rectángulo, un rectángulo que consiste en cuatro medidas computadas; si no, según lo especificado

La propiedad 'clip' se aplica solamente a los elementos posicionados absolutamente. Los valores tienen los significados siguientes:

auto

El elemento no se recorta.

< forma >

En CSS 2.1, el único valor válido de <forma> está: rect(<top>, <right>, <bottom>, <left>) donde <top> y <bottom> especifican los desplazamientos del límite del borde superior de la caja y <right>, y <left> especifican los desplazamientos del límite del borde izquierdo de la caja en texto de izquierda a derecha y del límite del borde derecho de la caja en texto de derecha a izquierda. Los autores deben separar valores de desplazamiento con comas. Las aplicaciones del usuario deben soportar la separación con comas, pero también pueden soportar la separación sin comas, porque una versión anterior de esta especificación era ambigua a este respecto.

<top>, <right>, <bottom> y <left> pueden tener cualquiera tener valor de <medida> o 'auto'. Se permiten las medidas negativas. El valor 'auto' significa que un borde dado de la zona de recorte será igual que el límite del borde de la caja generada del elemento (es decir, 'auto' significa igual que '0' para <top> y <left> (en texto de izquierda a derecha, <right> en texto de derecha a izquierda), igual que el valor computado de la altura más la suma de las anchuras verticales del relleno y del borde para <bottom>, e igual que el valor computado de la anchura más la suma de los anchos horizontales del relleno y del borde para <right> (en texto de izquierda a derecha, <left> en texto derecha a izquierda), así que cuatro valores 'auto' causan que la zona de recorte sea igual al borde de la caja del elemento).

Cuando las coordenadas se redondean a los coordenadas en píxel, se debe tener cuidado de que ningún píxel permanezca visible cuando <left> y <right> tengan el mismo valor (o <top> y <bottom> tengan el mismo valor), y recíprocamente que ningún píxel permanezca oculto dentro del borde de la caja del elemento cuando estos valores son 'auto'.

Una zona de recorte de un elemento recortado fuera de cualquier aspecto del elemento (por ejemplo, contenido, hijos, fondos, bordes, decoración del texto, contorno y mecanismos de desplazamiento visibles – si lo hay) que está fuera de la zona de recorte.

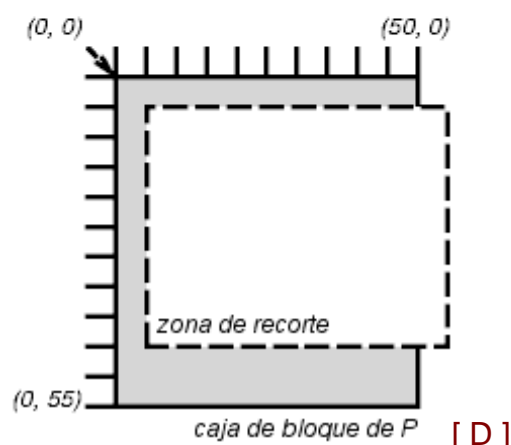
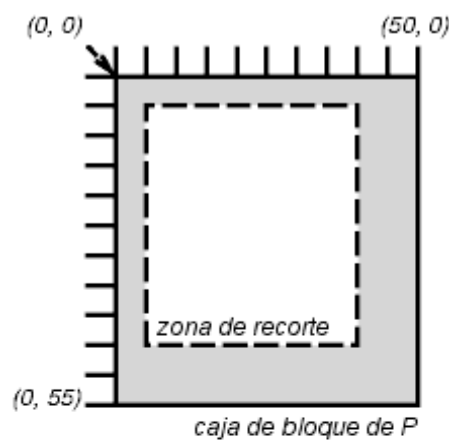
Los ascendentes del elemento pueden también recortar porciones de su contenido (por ejemplo, vía sus propias propiedades 'clip' y/o si la propiedad 'overflow' no es 'visible'); lo que se procesa es la intersección de las distintas zonas de recorte.

Si la zona de recorte excede los límites de la ventana del documento de la AU, el contenido se puede recortar a esa ventana por el entorno operativo nativo.

Las dos reglas siguientes:

```
p { clip: rect(5px, 40px, 45px, 5px); }  
p { clip: rect(5px, 55px, 45px, 5px); }
```

creará las zonas de recorte rectangulares delimitadas por las líneas discontinuas en las ilustraciones siguientes:



Nota. En CSS 2.1, todas las zonas de recorte son rectangulares. Anticipamos las extensiones futuras para permitir recortes no rectangulares. Las versiones futuras pueden también reintroducir una sintaxis para desplazar las formas de cada borde en vez del desplazamiento de un punto.

11.2 Visibilidad: la propiedad 'visibility'

'visibilidad'

Valor:	visible hidden collapse <u>inherit</u>
Inicial:	visible
Se aplica a:	todos los elementos
Se hereda:	sí
Porcentajes:	N/A
Medios:	<u>visual</u>
Valor computado:	según el especificado

La propiedad 'visibility' especifica si las cajas generadas por un elemento son procesadas. Las cajas invisibles siguen afectando la composición (poner la propiedad 'display' como 'none' para suprimir completamente la generación de la caja). Los valores tienen los significados siguientes:

visible

La caja generada es visible.

hidden

La caja generada es invisible (completamente transparente, no se dibuja nada), pero sigue afectando la composición. Además, los descendientes del elemento serán visibles si tienen 'visibility: visible'.

collapse

Consulte por favor la sección en [efectos dinámicos en filas y columnas](#) en las tablas. Si se usa en otros elementos que no sean filas, grupos de filas, columnas o grupos de columnas, 'collapse' tiene el mismo significado que 'hidden'.

Esta propiedad se puede utilizar conjuntamente con scripts para crear efectos dinámicos.

En el ejemplo siguiente, presionando cualquier botón del formulario se invoca una función script definida por el autor que hace que la correspondiente caja se haga visible y la otra se oculte. Puesto que estas cajas tienen el mismo tamaño y posición, el efecto es que una reemplaza a la otra (el código del script está en un lenguaje de script hipotético. Puede o no tener algún efecto en una AU apta para CSS):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/strict.dtd">
<HTML>
<HEAD><TITLE>Ejemplo de visibilidad dinámica</TITLE>
<META
http-equiv="Content-Script-Type"
content="application/x-hypothetical-scripting-language">
<STYLE type="text/css">
<!--
#container1 { position: absolute;
top: 2in; left: 2in; width: 2in }
#container2 { position: absolute;
top: 2in; left: 2in; width: 2in;
visibility: hidden; }
-->
</STYLE>
</HEAD>
<BODY>
<P>Elija un sospechoso:</P>
<DIV id="container1">
<IMG alt="Al Capone"
width="100" height="100"
src="suspect1.png">
<P>Nombre: Al Capone</P>
<P>Residencia: Chicago</P>
</DIV>

<DIV id="container2">
<IMG alt="Lucky Luciano"
width="100" height="100"
src="suspect2.png">
<P>Nombre: Lucky Luciano</P>
<P>Residencia: New York</P>
</DIV>

<FORM method="post"
action="http://www.suspect.org/process-bums">
<P>
<INPUT name="Capone" type="button"
value="Capone"
onclick='show("container1");hide("container2")'>
<INPUT name="Luciano" type="button"
value="Luciano"
onclick='show("container2");hide("container1")'>
</FORM>
</BODY>
</HTML>
```

12 Contenido generado, numeración automática y listas

Contenido

- [12.1 Los pseudo-elementos :before y :after](#)
- [12.2 La propiedad 'content'](#)
- [12.3 Comillas](#)
 - [12.3.1 Especificación de comillas con la propiedad 'quotes'](#)
 - [12.3.2 Inserción de comillas con la propiedad 'content'](#)
- [12.4 Contadores y numeración automática](#)
 - [12.4.1 Contadores anidados y área de alcance](#)
 - [12.4.2 Estilos de contadores](#)
 - [12.4.3 Contadores en elementos con 'display: none'](#)
- [12.5 Listas](#)
 - [12.5.1 Listas: las propiedades 'list-style-type', 'list-style-image', 'list-style-position' y 'list-style'](#)

En algunos casos, los autores pueden querer que las aplicaciones del usuario procesen contenidos que no vienen en la [estructura del documento](#). Un ejemplo familiar de esto es una lista numerada; el autor no quiere enumerar los números explícitamente, él o ella quiere que la aplicación del usuario la genere automáticamente. De modo similar, los autores pueden querer que la aplicación del usuario inserte la palabra "figura" antes del encabezado de una imagen, o "capítulo 7" antes del séptimo título del capítulo. Para audio o braille en particular, las aplicaciones del usuario deben poder insertar estas cadenas.

En CSS 2.1, el contenido se puede generar por dos mecanismos:

- La propiedad '[content](#)', junto con los pseudo-elementos :before y :after.
- Los elementos con un valor de 'list-item' para la propiedad '[display](#)'.

12.1 Los pseudo-elementos :before y :after

Los autores especifican el estilo y la ubicación del contenido generado con los pseudo-elementos :before y :after. Como sus nombres indican, los pseudo-elementos :before (antes) y :after (después) especifican la ubicación del contenido antes y después del contenido de la [estructura del documento](#) de un elemento. La propiedad '[content](#)', junto con estos pseudo-elementos, especifica lo que se inserta.

Por ejemplo, la regla siguiente inserta la cadena "Nota:" antes del contenido de cada elemento P cuyo atributo "class" tenga el valor "nota":

```
p.nota:before { content: "Nota:  " }
```

El formato de los objetos (ej., cajas) generados por un elemento incluye el contenido generado. Así pues, por ejemplo, cambiando la hoja de estilo anterior a:

```
p.nota:before { content: "Nota:  " }
p.nota { border: solid green }
```

causaría que un borde verde sólido fuera procesado alrededor de todo el párrafo, incluyendo la cadena inicial.

Los pseudo-elementos `:before` y `:after` [heredan](#) cualquier propiedad que se puede heredar del elemento en la estructura del documento al cual están ligados.

Por ejemplo, las reglas siguientes insertan comillas de apertura antes de cada elemento `Q`. El color de las comillas será rojo, pero la fuente será la misma que la del resto del elemento `Q`:

```
q:before {
  content: open-quote;
  color: rojo
}
```

En la declaración del pseudo-elemento `:before` o `:after`, las propiedades no-heredadas toman sus [valores iniciales](#).

Así pues, por ejemplo, debido a que el valor inicial de la propiedad `'display'` es `'inline'`, las comillas en el ejemplo previo se insertan como una caja a nivel de línea (es decir, en la misma línea que el contenido del texto inicial del elemento). El ejemplo siguiente fija explícitamente la propiedad `'display'` como `'block'`, de modo que el texto insertado se convierta en un bloque:

```
body:after {
  content: "Fin";
  display: block;
  margin-top: 2em;
  text-align: center;
}
```

Los elementos de los pseudo-elementos `:before` y `:after` interactúan con otras cajas, tales como cajas `run-in`, como si fueran realmente elementos insertado justamente dentro de su elemento asociado.

Por ejemplo, el fragmento de documento y la hoja de estilo el stylesheet siguientes:

```
< h2 > Cabecera </h2 >      h2 { display: run-in; }
< p > Texto </p >           p:before {display:block; content:'algun';}
```

... lo procesará exactamente de la misma manera que el fragmento del documento y hoja de estilo siguientes:

```
< h2 >Cabecera</h2 >      h2 { display: run-in; }
<p><span>Algún</span> Texto </p>  span{ display:block; }
```

Semejante, al fragmento de documento y hoja de estilo:

```
< h2 >Cabecera </h2 >      h2 { display:run-in; }
                               h2:after { display: block; content: 'cosa' }
< p > Texto </p >
```

... lo procesaría exactamente de la misma manera que el fragmento de documento y hoja de estilo siguientes:

```
< h2 > Cabecera < span>Cosa</span></h2 >
<p> Texto </p>
```

```
h2 { display: block; }
span {display: block; }
```

12.2 La propiedad **'content'**

'content'

<i>Valor:</i>	normal [<cadena> <uri> <contador> attr(<identificador>) open-quote close-quote no-open-quote no-close-quote] + <u>inherit</u>
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	pseudo-elementos :before y :after
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medios:</i>	<u>todos</u>
<i>Valor computado:</i>	para los valores de URI, el URI absoluto; para los valores de attr(), la cadena resultante; si no según el especificado

Esta propiedad se utiliza con los pseudo-elementos :before y :after para generar el contenido en un documento. Los valores tienen los siguientes significados:

normal

El pseudo-elemento no se genera.

< cadena >

Contenido del texto (véase la sección [Strings](#)).

< uri >

El valor es un URI que señala un recurso externo. Si una aplicación del usuario no puede soportar el recurso debido a los [tipos de medios](#) que soporta, debe ignorar el recurso.

< contador >

Los [contadores](#) se pueden especificar con dos funciones distintas: 'counter()' o 'counters()'. La primera función tienen dos formas: 'counter(nombre)' o 'counter(nombre, estilo)'. El texto generado es el valor del contador nombrado a este punto en la estructura del formato; se ajusta al formato del [estilo](#) indicado ('decimal' por defecto). La última función también tiene dos formas: 'counters(nombre, cadena)' o 'counters(nombre, cadena, estilo)'. El texto generado es el valor de todos los contadores con el nombre dado a este punto en la estructura del formato, separado por la cadena especificada. Los contadores se procesan con el [estilo](#) indicado ('decimal' por defecto). Vea la sección en [contadores y la numeración automática](#) para más información.

open-quote y close-quote

Estos valores son reemplazados por la cadena apropiada de la propiedad **'quotes'**.

no-open-quote y no-close-quote

Igual que 'none', pero incrementa (disminuye) el nivel de la anidamiento de las comillas.

attr(X)

Esta función devuelve como cadena el valor del atributo X para sujeto del selector. La cadena no es analizada por el procesador de CSS. Si el sujeto del selector no tiene un atributo X, se vuelve una cadena vacía. La distinción entre mayúsculas y minúsculas en el nombre de los atributos depende del lenguaje del documento. **Nota.** En CSS 2.1, no es posible referirse a los valores del atributo de otros elementos del sujeto del selector.

La propiedad **'display'** controla si el contenido está puesto en una caja de bloque, en línea, o marcador.

La regla siguiente hace que la cadena “Capítulo” sea generada antes que cada H1:

```
H1:before {
  contenido: "Capítulo: ";
  display: inline;
}
```

Los autores pueden incluir nuevas líneas en el contenido generado escribiendo la secuencia de escape “\A” en una de las cadenas después de la propiedad 'content'. Este salto de línea insertado está sujeto a la propiedad 'white-space' (espacio en blanco). Vea "[Cadenas](#)" y "[caracteres y mayúsculas/minúsculas](#)". para más información sobre la secuencia de escape “\A”.

```
h1:before {
  display: block;
  text-align: center;
  white-space: pre;
  content: "capítulo\A hoofdstuk\A chapitre"
}
```

El contenido generado no altera la estructura del documento. En particular, no retroalimenta el procesador del lenguaje del documento (ej., para otro análisis).

12.3 Comillas

En CSS 2.1, los autores pueden especificar, de un modo sensible al estilo y dependiente del contexto, como las aplicaciones del usuario deben procesar las comillas. La propiedad 'quotes' especifica pares de comillas cita para cada nivel de comillas incrustada. La propiedad 'content' da acceso a esas comillas y provoca que sean insertadas antes y después de una cita.

12.3.1 Especificación de comillas con la propiedad 'quotes'

'quotes'

Valor: [<cadena> <cadena>] +|none|inherit
Inicial: depende de la aplicación del usuario
Se aplica a: todos los elementos
Se hereda: sí
Porcentajes: N/A
Medios: visual
Valor computado: según el especificado

Esta propiedad especifica las comillas para cualquier cantidad de citas incrustadas. Los valores tienen los significados siguientes:

none

Los valores 'open-quote' y 'close-quote' de la propiedad 'content' no provocan ninguna comilla.

[<cadena> <cadena>] +

Los valores para los valores de 'open-quote' y 'close-quote' de la propiedad 'content' se toma de esta lista de pares de comillas (de apertura y de cierre). El primer par (más a la izquierda izquierdo) representa el nivel exterior de la cita, el segundo par el primer nivel de incrustación, etc. La aplicación del usuario debe aplicar el par de comillas apropiado según el nivel de incrustación.

Por ejemplo, aplicando la siguiente hoja de estilo:

```
/* Especifica los pares de comillas para dos niveles en dos idiomas */
q:lang(en) { quotes: ' ' ' ' ' ' ' ' }
q:lang(no) { quotes: "«" "»" ' ' ' ' }

/* Inserta las comillas antes y después del contenido del elemento Q */
q:before { content: open-quote }
q:after { content: close-quote }
```

al siguiente fragmento de HTML:

```
<HTML lang="en">
  <HEAD>
    <TITLE>Quotes</TITLE>
  </HEAD>
  <BODY>
    <P><Q>Quote me!</Q>
  </BODY>
</HTML>
```

permitiría que una aplicación del usuario produjera:

```
"Quote me!"
```

mientras que este fragmento de HTML:

```
<HTML lang="no">
  <HEAD>
    <TITLE>Quotes</TITLE>
  </HEAD>
  <BODY>
    <P><Q>Trøndere gråter når <Q>Vinsjan på kaia</Q> blir deklamert.</Q>
  </BODY>
</HTML>
```

produciría:

```
«Trøndere gråter når "Vinsjan på kaia" blir deklamert.»
```

Nota. Mientras que las comillas especificadas por 'quotes' en los ejemplos anteriores están situados convenientemente en los teclados de los ordenadores, las componedoras de texto de alta calidad requerirán diferentes caracteres de la ISO 10646. La siguiente tabla informativa enumera algunos de los caracteres de comillas de la ISO 10646:

Carácter	procesamiento aproximado	Código de la ISO 10646 (hex)	Descripción
"	"	0022	COMILLAS [comillas dobles en ASCII]
'	'	0027	APÓSTROFE [comilla simple en ASCII]
<	<	2039	COMILLA ANGULAR SIMPLE HACIA LA IZQUIERDA
>	>	20Á	COMILLA ANGULAR SIMPLE HACIA LA DERECHA
«	«	00AB	COMILLAS ANGULARES DOBLES HACIA LA IZQUIERDA
»	»	00BB	COMILLAS ANGULARES DOBLES HACIA LA DERECHA
‘	`	2018	COMILLA DE APERTURA SIMPLE [simple alta-6]
’	'	2019	COMILLA DE CIERRE SIMPLE [simple alta-9]
“	``	20Ç	COMILLAS DE APERTURA DOBLES [doble alta-6]
”	''	201D	COMILLAS DE CIERRE DOBLES [doble alta-9]
”	”	201E	COMILLAS DOBLES BAJA-9 [doble baja-9]

12.3.2 Inserción de comillas con la propiedad 'content'

Las comillas son insertadas en los lugares correspondientes de un documento con los valores 'open-quote' y 'close-quote' de la propiedad 'content'. Cada aparición de 'open-quote' o 'close-quote' es substituido por una de las cadenas del valor de 'quotes', en base a la profundidad del anidado.

'Open-quote' se refiere a la primera de un par de comillas, 'close-quote' se refiere a la segunda. Qué par de comillas se utiliza depende del nivel de anidamiento de las comillas: el número de apariciones de 'open-quote' en todo el texto generado antes de la presente, menos el número de apariciones de 'close-quote'. Si la profundidad es 0, se utiliza el primer par, si la profundidad es 1, se utiliza el segundo par, etc. Si la profundidad es mayor que el número de pares, se repite el ultimo par. Una 'close-quote' que se hiciera de profundidad negativa sería un error y es ignorada (en tiempo de proceso): la profundidad permanece en 0 y no se procesa rinde ninguna comilla (aunque el resto del valor de la propiedad 'content' todavía se inserta).

Nota. La profundidad de las comillas es independiente del anidamiento del documento fuente o de la estructura del formato.

Algunos estilos tipográficos requieren comillas de apertura que se repiten antes de cada párrafo de una cita que abarca varios párrafos, pero solo el último párrafo termina con comillas de cierre. En CSS, esto puede lograrse insertando comillas de cierre "fantasma". La palabra clave 'no-close-quote' disminuye el nivel de las citas pero no inserta una comilla.

La siguiente hoja de estilo pone comillas de apertura en cada párrafo en un BLOCKQUOTE, e inserta una sola comilla simple al final:

```
blockquote p:before { content: open-quote }
blockquote p:after  { content: no-close-quote }
blockquote p.last:after { content: close-quote }
```

Esto cuenta con que el último párrafo sea marcado con una clase "last".

Simétricamente, hay también una clave 'no-open-quote', que no inserta nada, pero incrementa en uno la profundidad de la cita.

12.4 Contadores y numeración automática

La numeración automática en CSS2 es controlada con dos propiedades, 'counter-increment' y 'counter-reset'. Los contadores definidos por estas propiedades son utilizados con las funciones counter() y counters() de la propiedad 'content'.

'counter-reset'

Valor: [<identificador> <entero>?]+ | none | inherit
inicial: none
Se aplica a: Todos los elementos
Se hereda: no
Porcentaje: N/A
Medio: todos
Valor computado: Como el especificado

'counter-increment'

Valor: [<identificador> <entero>?]+ | none | inherit
Inicial: none
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medios: todos
Valor computado: Como el especificado

La propiedad 'counter-increment' acepta uno o mas nombres de contadores (identificadores), cada uno seguido opcionalmente por un entero. El entero indica en cuanto se incrementa el contador con cada aparición del elemento. El incremento por defecto es 1. Los enteros cero y negativo son admitidos.

La propiedad 'counter-reset' también contiene una lista de uno o más nombres de contadores, cada uno seguido opcionalmente por un entero. El entero da el valor que en el contador es colocado con cada aparición del elemento. El valor por defecto es 0.

Si 'counter-increment' se refiere a un contador que no está en el [área de alcance \(ver abajo\)](#) de algún 'counter-reset', se asume que el contador ha sido reseteado a cero por el elemento raíz.

Este ejemplo muestra una forma de numerar capítulos y secciones con "Capítulo 1", "1.1", "1.2", etc.

```
H1:before {
  content: "Capítulo " counter(capítulo) ". ";
  counter-increment: capítulo; /* Añade 1 a capítulo */
  counter-reset: Sección;      /* Pone la sección a 0 */
}
H2:before {
  content: counter(capítulo) "." counter(Sección) " ";
  counter-increment: Sección;
}
```

Si un elemento incrementa/resetea un contador y también lo usa (en la propiedad 'content' de su pseudo elemento :before o :after), el contador se utiliza después de ser incrementado/reseteado.

Si un elemento resetea e incrementa un contador, el contador es reseteado primero y luego incrementado.

La propiedad 'counter-reset' sigue las reglas de cascada. De modo que, debido a la cascada, la siguiente hoja de estilo:

```
H1 { counter-reset: Sección -1 }
H1 { counter-reset: imagenum 99 }
```

Solamente reseteará 'imagenum'. Para resetear ambos contadores, estos tienen que ser especificados a la vez:

```
H1 { counter-reset: Sección -1 imagenum 99 }
```

12.4.1 Contadores anidados y área de alcance

Los contadores son "autoanidados", en el sentido de que la re-utilización de un contador en un elemento hijo automáticamente crea una nueva instancia del contador. Esto es importante en situaciones como las listas den HTML, donde los elementos pueden ser anidados dentro de si mismo hasta una profundidad arbitraria. Resultaría imposible definir los contadores con un nombre único para cada nivel.

Así, lo siguiente es suficiente para numerar los elementos (ítems) de una lista anidada. El resultado es muy similar al de poner 'display:list-item' y 'list-style: inside' en el elemento LI:

```
OL { counter-reset: item }
LI { display: block }
LI:before { content: counter(item) ". "; counter-increment: item }
```

El autoanidamiento se basa en el principio de que cada elemento que tiene 'counter-reset' para un contador X, crea un nuevo contador X, el área de alcance del cual es el elemento, sus hermanos siguientes y todos los descendientes del elemento y sus siguientes hermanos.

En el ejemplo de arriba, un OL creará un contador, y todos los hijos de OL se referirán a ese contador.

Si señalamos con `item[n]` la n^o instancia del contador "item" y con "(" y ")" el principio y final del área de alcance, entonces el siguiente fragmento de HTML utilizará los contadores indicados. (Asumimos una hoja de estilo como la proporcionada en el ejemplo de arriba).

```
<OL>          <!-- (pone item[0] a 0          -->
  <LI>item     <!-- incrementa item[0] (= 1) -->
  <LI>item     <!-- incrementa item[0] (= 2) -->
    <OL>       <!-- (pone item[1] a 0          -->
      <LI>item  <!-- incrementa item[1] (= 1) -->
      <LI>item  <!-- incrementa item[1] (= 2) -->
      <LI>item  <!-- incrementa item[1] (= 3) -->
        <OL>   <!-- (pone item[2] a 0          -->
          <LI>item <!-- incrementa item[2] (= 1) -->
        </OL>   <!-- )                          -->
      <OL>     <!-- (pone item[3] a 0          -->
        <LI>    <!-- incrementa item[3] (= 1) -->
      </OL>    <!-- )                          -->
    <LI>item   <!-- incrementa item[1] (= 4) -->
  </OL>       <!-- )                          -->
  <LI>item     <!-- incrementa item[0] (= 3) -->
  <LI>item     <!-- incrementa item[0] (= 4) -->
</OL>        <!-- )                          -->
<OL>         <!-- (resetea item[4] a 0        -->
  <LI>item     <!-- incrementa item[4] (= 1) -->
  <LI>item     <!-- incrementa item[4] (= 2) -->
</OL>        <!-- )                          -->
```

La función '`counters()`' genera una cadena compuesta por los valores de todos los contadores con el mismo nombre, esperados por una cadena dada.

La siguiente hoja de estilos numera los elementos (ítems) de una lista como "1", "1.1", "1.1.1", etc.

```
OL { counter-reset: item }
LI { display: block }
LI:before { content: counters(item, "."); counter-increment: item }
```

12.4.2 Estilos de contadores

Por defecto, los contadores son procesados con números decimales, pero todos los estilos disponibles para la propiedad 'list-style-type' están también disponibles para los contadores. La sintaxis es:

```
counter(nombre)
```

para el estilo por defecto, o :

```
counter(nombre, 'list-style-type')
```

Todos los estilos están permitidos, incluyendo 'disc', 'circle', 'square', y 'none'.

```
H1:before      { content: counter(chno, upper-latin) ". " }
H2:before      { content: counter(Sección, upper-roman) " - " }
BLOCKQUOTE:after { content: " [" counter(bq, hebrew) "]" }
DIV.note:before { content: counter(notecntr, disc) " " }
P:before       { content: counter(p, none) }
```

12.4.3 Contadores en elementos con 'display: none'

Un elemento que no es mostrado ('display' fijado a 'none') no puede incrementar o resetear un contador.

Por ejemplo, con la siguiente hoja de estilo, los H2 de la clase (class) "secreto" no incrementan a 'count2'.

```
H2.secreto {counter-increment: count2; display: none}
```

Los elementos con 'visibility' puesto a 'hidden', por otro lado, si incrementan los contadores.

12.5 Listas

CSS 2.1 ofrece un formato básico visual de listas. Un elemento con 'display: list-item' genera una [caja principal](#) para el contenido de los elementos y un marcador opcional de caja como una indicación visual de que el elemento es una lista de objetos.

Las propiedades de lista describen un formato visual básico de listas: estas permiten a hojas de estilo especificar el tipo de marcador (imagen, glyph, o numero), y la posición del marcador con respecto a la caja principal (fuera de ella o dentro de ella antes del contenido). No permiten a los autores especificar distintos estilos (colores, fuentes, alineación, etc) para la lista de marcadores o ajustar su posición con respecto a la caja principal, estos pueden ser derivados de la caja principal.

Las [propiedades del fondo](#) se aplican solo a la caja principal, un marcador de caja 'outside' es transparente.

12.5.1 Listas: las propiedades 'list-style-type', 'list-style-image', 'list-style-position', y 'list-style'

'list-style-type'

<i>Valor:</i>	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian none <u>inherit</u>
<i>Inicial:</i>	disc
<i>Se aplica a:</i>	Elementos con 'display: list-item'
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

Esta propiedad especifica la apariencia del marcador de los elementos (items) de la lista si 'list-style-image' tiene el valor 'none' o si la imagen señalada por el URI no puede ser mostrada. El valor 'none' especifica ningún marcador, de lo contrario hay 3 tipos de marcadores: glyphs, sistema numérico, y sistema alfabético.

Los Glyphs son especificados con **disc**, **circle**, y **square**. Su procesamiento exacto depende de la aplicación del usuario

Los sistemas numéricos son especificados con:

decimal

Número decimal, comenzando con 1.

decimal-leading-zero

Números decimales complementarios con ceros iniciales (ej., 01, 02, 03, ..., 98, 99).

lower-roman

Números romanos en minúsculas (i, ii, iii, iv, v, etc.).

upper-roman

Números romanos en mayúsculas (I, II, III, IV, V, etc.).

georgian

Numeración georgiana tradicional (an, ban, gan, ..., he, tan, in, in-an, ...).

armenian

Numeración armenia tradicional.

Una aplicación del usuario que no reconoce un sistema numérico debe utilizar el 'decimal'.

Los sistemas alfabéticos son especificados con:

lower-latin or lower-alpha

Letras minúsculas en ascii (a, b, c, ... z).

upper-latin or upper-alpha

Letras mayúsculas en ascii (A, B, C, ... Z).

lower-greek

Minúsculas griegas clásicas alpha, beta, gamma, ... (α , β , γ , ...)

Esta especificación no define como los sistemas alfabéticos se desenvuelven al final del alfabeto. Por ejemplo, después de 26 elementos (items) de una lista, el procesamiento de 'lower-latin' es indefinido. Por eso, para listas extensas, recomendamos que los autores especifiquen números reales.

Por ejemplo, el siguiente documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Numeración con latinas minúsculas</TITLE>
    <STYLE type="text/css">
      ol { list-style-type: lower-roman }
    </STYLE>
  </HEAD>
  <BODY>
    <OL>
      <LI> Este es el primer elemento.
      <LI> Este es el segundo elemento.
      <LI> Este es el tercer elemento.
    </OL>
  </BODY>
</HTML>
```

Puede producir algo así:

```
i Este es el primer elemento.
ii Este es el segundo elemento.
iii Este es el tercer elemento.
```

La alineación de los marcadores de la lista (aquí, justificados a la derecha) depende de la aplicación del usuario.

'list-style-image'

Valor: <uri> | none | inherit
Inicial: none
Se aplica a: Elementos con 'display: list-item'
Se hereda: Si
Porcentajes: N/A
Medio: visual
Valor computado: URI absoluta o 'none'

Esta propiedad define la imagen que será utilizada como marcador de elementos (items) de una lista. Cuando la imagen está disponible, reemplazará el marcador definido con el marcador 'list-style-type'.

El siguiente ejemplo define que el marcador al comienzo de cada elemento (item) de una lista se la imagen "ellipse.png".

```
ul { list-style-image: url("http://png.com/ellipse.png") }
```


'list-style-position'

<i>Valor:</i>	inside outside <u>inherit</u>
<i>Inicial:</i>	outside
<i>Se aplica a:</i>	Elementos con 'display: list-item'
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

Esta propiedad especifica la posición de la caja marcador en la caja de bloque principal. Los valores tienen los siguientes significados:

outside

La caja del marcador está fuera de la caja de bloque principal. CSS2.1 no especifica la localización precisa de la caja de marcador.

inside

La caja del marcador está en la primera caja en línea en la caja de bloque principal, después de la cual fluye el contenido del elemento. CSS2.1 no especifica la localización precisa para la caja del marcador.

Por ejemplo:

```
<HTML>
  <HEAD>
    <TITLE>Comparación entre posición inside/outside</TITLE>
    <STYLE type="text/css">
      ul          { list-style: outside }
      ul.compact { list-style: inside }
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>el primer elemento de la lista viene primero
      <LI>el segundo elemento de la lista viene segundo
    </UL>

    <UL class="compact">
      <LI>el primer elemento de la lista viene primero
      <LI>el segundo elemento de la lista viene segundo
    </UL>
  </BODY>
</HTML>
```

El ejemplo de arriba puede ser procesado como:

- el primer ítem de la lista viene primero
- el segundo ítem de la lista viene después

-
- el primer ítem de la lista viene primero
 - el segundo ítem de la lista viene después

↑
 Los costados izquierdos de las cajas de los ítems de la lista no son afectados por la ubicación de los marcadores.

[D]

En un texto de derecha a izquierda, los marcadores habrían estado al costado derecho de la caja.

'list-style'

Valor: [<'list-style-type'> || <'list-style-position'> || <'list-style-image'>] | inherit
Inicial: Ver propiedades individuales
Se aplica a: Elementos con with 'display: list-item'
Se hereda: Si
Porcentajes: N/A
Medio: visual
Valor computado: Ver propiedades individuales

La propiedad 'list-style' es una notación corta para definir las tres propiedades 'list-style-type', 'list-style-image', y 'list-style-position' en el mismo lugar en la hoja de estilo.

```
ul { list-style: upper-roman inside } /* cualquier elemento "ul" */
ul > li > ul { list-style: circle outside } /* cualquier "ul" hijo
de un "li" hijo
de un elemento "ul" */
```

Aunque los autores pueden especificar la información sobre 'list-style' directamente en los elementos ítem de una lista (ej., "li" en HTML), deben hacerlo con cuidado. Las siguientes reglas parecen similares, pero la primera declara un selector descendente y la segunda un (más específico) selector hijo.

```
ol.alpha li { list-style: lower-alpha } /* cualquier "li" descendente de
un "ol" */
ol.alpha > li { list-style: lower-alpha } /* cualquier "li" hijo de un
"ol" */
```

Los autores que utilizan solo el selector de descendente pueden no lograr los resultados que esperan. Considere las siguientes reglas:

```

<HTML>
  <HEAD>
    <TITLE>ABISO: Resultados inesperados debido a la cascada</TITLE>
    <STYLE type="text/css">
      ol.alpha li { list-style: lower-alpha }
      ul li      { list-style: disc }
    </STYLE>
  </HEAD>
  <BODY>
    <OL class="alpha">
      <LI>nivel 1
      <UL>
        <LI>nivel 2
      </UL>
    </OL>
  </BODY>
</HTML>

```

El procesamiento deseado tendría los elementos de la lista del nivel 1 con etiquetas 'lower-alpha' y los elementos del nivel 2 con etiquetas 'disc'. Sin embargo, el [orden de cascada](#) provocará que la primera regla de estilo (que incluye información de la clase (class) específica) enmascare la segunda. Las siguientes reglas solucionan el problema empleando en cambio un [selector hijo](#):

```

ol.alpha > li { list-style: lower-alpha }
ul li      { list-style: disc }

```

Otra solución sería especificar la información sobre 'list-style' solo en los elementos de tipo lista:

```

ol.alpha { list-style: lower-alpha }
ul      { list-style: disc }

```

La herencia transferirá los valores de 'list-style' de los elementos OL y UL a los elementos LI. Este es el camino recomendado para especificar la información acerca del estilo de lista.

Un valor URI puede combinarse con cualquier otro valor, como en:

```

ul { list-style: url("http://png.com/ellipse.png") disc }

```

En el ejemplo de arriba, 'disc' será utilizado cuando la imagen no esté disponible..

Un valor de 'none' para la propiedad 'list-style' pone a 'list-style-type' y 'list-style-image' como 'none':

```

ul { list-style: none }

```

El resultado es que ningún marcador de elementos de lista es mostrado.

13 Medios paginados

Contenido

- [13.1 Introducción a los medios paginados](#)
- [13.2 Cajas de página: la regla @page](#)
 - [13.2.1 Márgenes de la página](#)
 - [Procesando las cajas de página que no ajustan a una hoja de destino](#)
 - [Ubicación de la caja de página en la hoja](#)
 - [13.2.2 Selectores de página: seleccionando paginas izquierda, derecha, y primera](#)
 - [13.2.3 Contenido fuera de caja de la página](#)
- [13.3 Saltos de página](#)
 - [13.3.1 Propiedades de salto de página: 'page-break-before', 'page-break-after' 'page-break-inside'](#)
 - [13.3.2 Saltos dentro de los elementos: 'orphans' y 'widows'](#)
 - [13.3.3 Saltos de páginas permitidos](#)
 - [13.3.4 Saltos de páginas forzados](#)
 - [13.3.5 Los "mejores" saltos de página](#)
- [13.4 Cascada en el contexto de la página](#)

13.1 Introducción a los medios paginados

Los medios paginados (ej., papel, transparencias, páginas que se muestran en la pantalla del ordenador, etc.) se diferencian de los [medios continuos](#) en que el contenido del documento está dividido en una o más páginas discretas. Para manejar las páginas, CSS 2.1 describe cómo los márgenes de la página se fijan en las [cajas de la página](#), y cómo se declaran los [saltos de página](#).

La aplicación del usuario es la responsable para transferir las cajas de pagina de un documento sobre las hojas reales donde el documento el documento será procesado en ultima instancia (papel, transparencia, pantalla, etc.). A menudo hay una relación 1-a-1 entre una caja de página y una hoja, pero éste no es siempre el caso. Las posibilidades de la transferencia incluyen:

- Transferencia de una caja de la página a una hoja (ej., impresión de una sola cara).
- Transferencia de dos cajas de la página a ambos lados de la misma hoja (ej., impresión a doble cara).
- Transferencias de N (pequeñas) cajas de la página una misma hoja (llamada "n-up").
- Transfiriendo una caja (grande) de la página a hojas de N x M (llamadas "mosaico").
- Crear firmas. Una firma es un grupo de páginas impresas en una hoja, que, cuando están dobladas y ajustadas como un libro, aparecen en su secuencia correcta.
- Impresión de un documento en varias bandejas de salida.
- Producir la salida de un archivo.

13.2 Cajas de página: la regla @page

La *caja de página* es una región rectangular que contiene dos áreas:

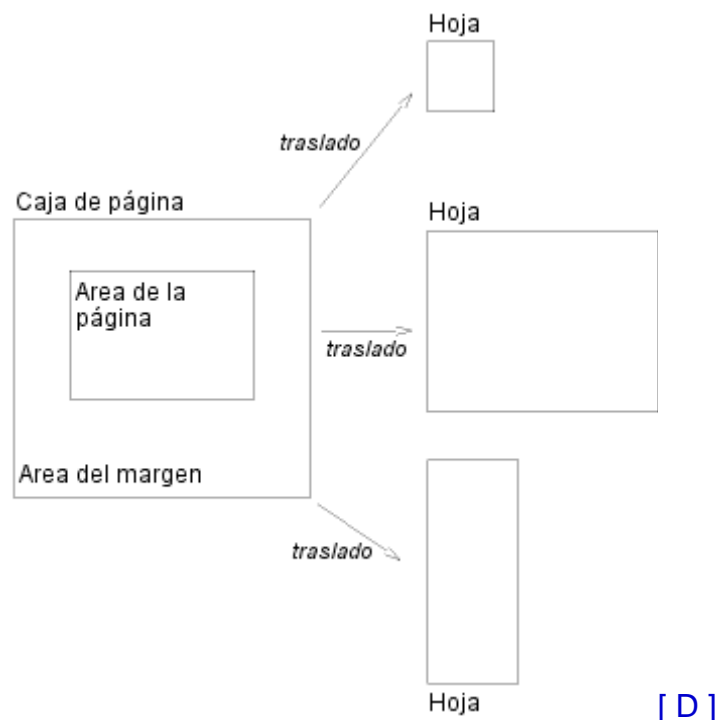
- *El área de la página* . El área de la página incluye las cajas presentadas en esa página. Los límites del área de la página actúan como el [bloque de contenido](#) inicial de la disposición efectuada entre saltos de página.
- El área del margen, que rodea al área de la página.

Los autores pueden especificar los márgenes de una caja de página dentro de una regla @page. Una regla @page consiste en la palabra clave "@page", seguido por un selector opcional de la página, seguido por un bloque de declaraciones. Las declaraciones en una regla @page serían en *el contexto de la página* .

El selector de la página especifica a qué páginas se aplican las declaraciones. En CSS 2.1, los selectores de páginas pueden señalar la primera página, todas las páginas izquierdas, o todas las páginas derechas

13.2.1 Márgenes de la página

En CSS 2.1, solamente las [propiedades del margen](#) ('margin-top', 'margin-right', 'margin-bottom', 'margin-left', y 'margin') se aplican dentro del [contexto de la página](#) . El siguiente diagrama demuestra las relaciones entre la hoja, la caja de página, y los márgenes de la página:



Aquí está un simple ejemplo que fija todos los márgenes de la página en todas las páginas:

```
@page { margin: 3cm; }
```

El [contexto de la página](#) no tiene ninguna noción de fuentes, así que las unidades 'em' y 'ex' no se permiten. Los valores de porcentajes en las propiedades del margen son relativas a las dimensiones de la [caja de página](#); para los márgenes izquierdo y derecho, refieren al ancho de la caja de página mientras que para los márgenes superior e inferior, refieren a la altura de la caja de página. Todas las demás unidades asociadas con las respectivas propiedades de CSS2.1 están permitidas.

Debido a los valores negativos del margen (en la caja de página o en elementos) o del [posicionamiento absoluto](#) el contenido puede terminar fuera de la caja de página, pero este contenido puede ser "cortado" -- por la aplicación del usuario, la impresora, o en última instancia, la guillotina de papel.

El valor computado de los márgenes de la caja para la parte superior o inferior del área de la página es cero.

Procesando las cajas de página que no caben una hoja de destino

Si una caja de página no se ajusta a las dimensiones de la hoja de destino, la aplicación del usuario puede elegir:

- Rotar 90° la caja de página si esto hace que la página se ajuste.
- Escalar la página para ajustarla al destino.

La aplicación del usuario debe consultar al usuario antes de realizar estas operaciones.

Ubicación de la caja de página en la hoja

Cuando la caja de página es más pequeña que el tamaño del destino, la aplicación del usuario es libre de ubicar la caja de página en cualquier lugar de la hoja. Sin embargo, se recomienda que la caja de página esté centrada en la hoja puesto que así se alinearán las páginas de doble cara y evitará la pérdida accidental de información que es impresa cerca del borde de la hoja.

13.2.2 Selectores de la página: seleccionando páginas izquierda, derecha, y primera

Al imprimir documentos de doble cara, la [cajas de página](#) en páginas izquierdas y derechas pueden ser diferentes. Esto se puede expresar a través de dos pseudo-classes de CSS que se puedan utilizar en selectores de la página.

Todas las páginas son clasificadas automáticamente por las aplicaciones del usuario en la pseudo-clase :left o :right.

```
@page :left {
  margin-left: 4cm;
  margin-right: 3cm;
}

@page :right {
  margin-left: 3cm;
  margin-right: 4cm;
}
```

Los autores también pueden especificar el estilo para la primera página de un documento con la pseudo-class `:first`:

```
@page { margin: 2cm } /* Todos los márgenes fijados a 2cm */

@page :first {
  margin-top: 10cm /* Margen superior de la primera página a 10cm */
}
```

Las propiedades especificadas en una regla `@page :left` o `:right` reemplaza las especificadas en una regla `@page` que no tiene especificada ninguna pseudo-clase. Las propiedades especificadas en una regla `@page :first` reemplaza las especificadas en las reglas `@page :left` o `:right`.

Las declaraciones de márgenes sobre paginas izquierda (`left`), derecha (`right`) y primera (`first`) pueden causar anchos diferentes en el [área de la página](#). Para simplificar las implementaciones, las aplicaciones del usuario pueden utilizar una simple anchura del área de la página en páginas izquierdas, derechas, y primeras. En este caso, la anchura del área de la página de la primera página debe ser utilizada.

13.2.3 Contenido fuera de la caja de la página

Al estructurar el contenido en el modelo de la página, parte del contenido puede terminar fuera de la caja de página. Por ejemplo, un elemento cuya propiedad `'white-space'` tiene el valor `'pre'` puede generar una caja que sea más ancha que la caja de página. También, cuando las cajas son posicionadas [absolutamente](#), pueden terminar en ubicaciones "inconveniente". Por ejemplo, las imágenes se pueden poner sobreborde de la caja de página o 100.000 metros por debajo de la caja de página.

El procesamiento exacto de tales elementos queda fuera del alcance de esta especificación. Sin embargo, recomendamos que los autores y las aplicaciones del usuario observen los principios generales siguientes referentes al contenido fuera de la caja de página:

- El contenido debe ser admitido ligeramente fuera de la caja de la página para permitir que "sangrado" de las páginas.
- Las aplicaciones del usuario deben evitar generar una gran cantidad de cajas de página vacías para respetar la ubicación de los elementos (ej., nadie quiere imprimir 100 páginas en blanco).
- Los autores no deben posicionar elementos en localizaciones inconvenientes para solo para evitar procesarlos.
- Las aplicaciones del usuario pueden manipular las cajas ubicadas fuera de la caja de página de varias maneras, incluyendo el descarte de las mismas o crear la cajas de página para ellas al final del documento.

13.3 Saltos de página

Esta sección describe los saltos de página en CSS 2.1. Cinco propiedades indican donde puede o debe la aplicación de usuario producir los saltos de página, y en qué página (izquierda o derecha) debe continuar el contenido subsecuente. Cada salto de página termina la composición de la [caja de página](#) actúa y provoca que las partes restantes de la [estructura del documento](#) sean colocadas en una nueva caja de página.

13.3.1 Propiedades de salto de página : **'page-break-before'**, **'page-break-after'**, **'page-break-inside'**

'page-break-before'

Valor: auto | always | avoid | left | right | inherit
Inicial: auto
Se aplica a: Elementos a nivel de bloque (block-level elements)
Se hereda: no
Porcentajes: N/A
Medio: visual, paginados
Valor computado: Como el especificado

'page-break-after'

Valor: auto | always | avoid | left | right | inherit
Inicial: auto
Se aplica a: Elementos a nivel de bloque (block-level elements)
Se hereda: no
Porcentajes: N/A
Medio: visual, paginados
Valor computado: Como el especificado

'page-break-inside'

Valor: avoid | auto | inherit
Inicial: auto
Se aplica a: Elementos a nivel de bloque (block-level elements)
Se hereda: Si
Porcentajes: N/A
Medio: visual, paginados
Valor computado: Como el especificado

Los valores para estas propiedades tienen los significados siguientes:

auto

Ni fuerza ni prohíbe un salto de página antes de (después de, en) la caja generada.

always

Siempre fuerza un salto de página antes (después) de la caja generada.

Avoid:

Evita un salto de página antes (después de, en) la caja generada.

left

Fuerza uno o dos saltos de páginas antes (después) de la caja generada para ajustar el formato la página siguiente como una página izquierda.

right

Fuerce uno o dos saltos de página antes (después) de la caja generada para ajustar el formato de la página siguiente como una página derecha.

Si la primera página de un documento es :left o :right depende de la dirección principal de la escritura del documento. Una aplicación del usuario con conformidad puede interpretar los valores 'left' y 'right' como 'always'.

Una localización potencial de un salto de página está típicamente bajo la influencia de la propiedad 'page-break-inside' del elemento padre, la propiedad 'page-break-after' del elemento precedente y la propiedad 'page-break-before' del elemento siguiente. Cuando estas propiedades tienen valores distintos de 'auto', los valores 'always', 'left', y 'right' tiene prioridad sobre 'avoid'.

Estas propiedades se aplican solamente a los elementos a nivel de bloque que están en el flujo normal del elemento raíz.

13.3.2 Saltos dentro de los elementos: 'orphans' y 'widows'

'orphans'

Valor: <entero> | inherit
Inicial: 2
Se aplica a: Elementos a nivel de bloque (block-level elements)
Se hereda: Si
Porcentajes: N/A
Medio: visual, paginados
Valor computado: Como el especificado

'widows'

Valor: <entero> | inherit
Inicial: 2
Se aplica a: Elementos a nivel de bloque (block-level elements)
Se hereda: Si
Porcentajes: N/A
Medio: visual, paginados
Valor computado: Como el especificado

La propiedad 'orphans' especifica el número mínimo de líneas de un párrafo que se deben dejar al final de una página. La propiedad 'widows' especifica el número mínimo de líneas de un párrafo que se deban dejar al comienzo de una página. Los ejemplos de cómo se utilizan para controlar los saltos de página se dan abajo.

Para información sobre el formato del párrafo, consulte por favor la sección sobre [cajas de línea](#).

13.3.3 Saltos de página permitidos

En el flujo normal, los saltos de página pueden ocurrir en los lugares siguientes:

1. En el margen vertical entre las cajas de bloque. Cuando ocurre un salto de página aquí, los [valores usados](#) de las propiedades relevantes '[margin-top](#)' y '[margin-bottom](#)' se fijan a '0'.
2. Entre [cajas de línea](#) dentro de una caja de [bloque](#).

Estos saltos están sujetos a las reglas siguientes:

- **Regla A:** Provocar un salto en (1) está permitido solo si las propiedades '[page-break-after](#)' y '[page-break-before](#)' de todos los elementos que generan cajas que se justan a este margen lo permiten, lo cual sucede cuando al menos un de ellos tiene el valor 'always', 'left', o 'right', o cuando todos son 'auto'.
- **Regla B:** Sin embargo, si todos son 'auto' y el ascendente común más cercano de todos los elementos tiene un valor para '[page-break-inside](#)' de 'avoid', entonces el salto aquí no está permitido.
- **Regla C:** Provocar un salto en (2) está permitido solo si el número de [cajas de línea](#) entre el salto y el comienzo de la caja de bloque que las encierra es el valor de '[orphans](#)' o más, y el número de cajas de línea entre el salto y el final de la caja es el valor de '[widows](#)' o más.
- **Regla D:** Además, provocar un salto en (2) está permitido solo si la propiedad '[page-break-inside](#)' es 'auto'.

Si lo anterior no proporciona suficientes puntos de ruptura para mantener al contenido fluyendo entre las cajas de páginas, entonces las reglas B y D se abandonan para encontrar puntos adicionales de saltos.

Si eso aún no conduce a una cantidad de puntos de ruptura suficientes, las reglas A y B son también abandonadas, para encontrar todavía más puntos para los saltos.

13.3.4 Saltos de página forzados

Un salto de página debe producirse en (1) si, entre las propiedades '[page-break-after](#)' y '[page-break-before](#)' de todos los elementos de las cajas generadas que se ajustan a ese margen, hay al menos una con el valor 'always', 'left', o 'right'.

13.3.5 Los "mejores" saltos de página

CSS2 no define cual de todo el conjunto de saltos de paginas permitidos debe utilizarse; CSS2 no prohíbe que la aplicación del usuario realice el salto en cada posible punto de corte, o que no lo haga en absoluto. Pero CSS2 recomienda que las aplicaciones del usuario observen las siguientes reglas heurísticas (mientras reconociendo que estas son a veces contradictorias):

- Producir los menores saltos posibles.
- Hacer que todas las paginas que no terminan con un salto forzado parezcan tener aproximadamente la misma altura.
- Evitar saltos dentro de un bloque que tiene un borde.

- Evitar saltos dentro de una tabla.
- Evitar saltos dentro de un elemento flotante

Suponga, por ejemplo, que la hoja de estilo contiene 'orphans: 4', 'widows: 2', y hay 20 líneas (cajas de línea) disponibles al final de la página actual:

- Si un párrafo al final de la página actual contiene 20 líneas o menos, debe colocarse en la página actual.
- Si un párrafo contiene 21 o 22 líneas, la segunda parte del párrafo no debe violar la restricción 'widows', y así la segunda parte debe contener exactamente dos líneas.
- Si el párrafo contiene 23 líneas o más, la primera parte debe contener 20 líneas y la segunda parte el resto de las líneas.

Ahora suponga que 'orphans' es '10', 'widows' es '20', y hay 8 líneas disponibles al final de la página actual:

- Si un párrafo al final de la página actual contiene 8 líneas o menos, debe colocarse en la página actual.
- Si el párrafo contiene 9 líneas o más, no puede ser cortado (esto violaría la restricción impuesta a las líneas orphan (huérfanas)), así que debe moverse como un bloque a la siguiente página.

13.4 Cascada en el contexto de página

Las declaraciones en el contexto de pagina obedecen la cascada tanto como las declaraciones normales de CSS2.

Considere el siguiente ejemplo: Consider the following example:

```
@page {  
  margin-left: 3cm;  
}  
  
@page :left {  
  margin-left: 4cm;  
}
```

Debido a la mayor especificidad del selector de la pseudo-clase, el margen izquierdo en las paginas de la izquierda será de '4cm' y todas las demás páginas (es decir, las páginas de la derecha) tendrán un margen izquierdo de '3cm'.

14 Colores y fondos

Contenido

- [14.1 Color del primer plano: la propiedad 'color'](#)
- [14.2 El fondo](#)
 - [14.2.1 Propiedades del fondo: 'background-color', 'background-image', 'background-repeat', 'background-attachment' 'background-position' y 'background'](#)
- [14.3 Corrección gamma](#)

Las propiedades de CSS permiten que los autores especifiquen el color del primer plano y el del fondo de un elemento. Los fondos pueden ser colores o imágenes. Las propiedades del fondo permiten que los autores colocar una imagen de fondo, repetirla, y declarar si debe quedar fija con respecto al [acceso visual](#) o se desliza junto con el documento.

Vea la sección sobre [unidades de color](#) para la sintaxis de los valores válidos del color.

14.1 Color del primer plano: la propiedad **'color'**

'color'

<i>Valor:</i>	<code><color></code> <code>inherit</code>
<i>Inicial:</i>	Depende de la ampliación del usuario
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<code>visual</code>
<i>Valor computado:</i>	Como el especificado

Esta propiedad describe el color del primer plano del contenido del texto de un elemento. Hay diversas maneras de especificar rojo:

```
em { color: rojo }/* predefinido */
em { color: rgb (255,0,0) }/* RGB rango 0-255 */
```

14.2 El fondo

Los autores pueden especificar el fondo de un elemento (es decir, su superficie de procesamiento) ya sea como un color o una imagen. En términos del [modelo de caja](#), el "background" refiere al fondo de las áreas del [contenido](#), de [relleno](#) y del [borde](#). Los colores y los estilos del borde se fijan con las [propiedades del borde](#). Los márgenes son siempre transparentes.

Las propiedades del fondo no se heredan, pero el fondo de la caja del padre por defecto se verá a través de la misma por defecto debido al valor inicial 'transparent' para `'background-color'`.

El fondo del elemento raíz se convierte en el fondo del lienzo y cubre todo el [lienzo](#), anclado en el mismo punto como si este fuese pintado solo para el elemento raíz. El elemento raíz no pinta este fondo otra vez.

Para los documentos del HTML, sin embargo, recomendamos que los autores especifiquen el fondo para el elemento del BODY en vez del elemento HTML. Las aplicaciones del usuario observar las siguientes reglas de precedencia para completar el fondo del lienzo de los documentos HTML: si el valor de la propiedad 'background' para el elemento HTML es distinto a 'transparent' entonces lo utiliza, de otro modo se usa el valor de la propiedad 'background' para el elemento BODY. Si el valor resultante es 'transparent', el procesamiento es indefinido. Esto no se aplica a los documentos de XHTML.

Según estas reglas, el lienzo subyacente del siguiente documento HTML tendrá un fondo "mármol":

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
  <TITLE>Poniendo el fondo del lienzo</TITLE>
  <STYLE type="text/css">
    BODY { background: url("http://ejemplo.com/marmol.png") }
  </STYLE>
  <P>Mi fondo es de mármol.
```

Observe que la regla para el elemento BODY funcionará aunque la etiqueta BODY se ha omitido en el código fuente del HTML puesto que el programa de análisis del HTML deducirá la etiqueta que falta.

Los fondos de los elementos que forman un contexto de apilamiento (véase la propiedad 'z-index') se pintan en el fondo del contexto de apilamiento del elemento, debajo de cualquier contenido del contexto de apilamiento.

14.2.1 Propiedades del fondo : 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position', y 'background'

'background-color'

<i>Valor:</i>	<u><color></u> transparent <u>inherit</u>
<i>Inicial:</i>	transparent
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

Esta propiedad fija el color de fondo de un elemento, ya sea un valor <color> o una palabra clave 'transparent', para hacer que los colores subyacentes se vean a través.

```
h1 { fondo-color: #F00 }
```

'background-image'

<i>Valor:</i>	<uri> none <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	URI absoluta

Esta propiedad fija la imagen de fondo de un elemento. Al fijar una imagen de fondo, los autores deben también especificar un color de fondo que será utilizado cuando la imagen no está disponible. Cuando la imagen está disponible, se procesa encima del color de fondo. (así, el color es visible a través de las partes transparentes de la imagen).

Los valores para esta propiedad son cualquier < uri >, para especificar la imagen, o 'none', cuando no se utiliza ninguna imagen.

```
BODY { background-image: url("mármol.gif") }
P { background-image: none }
```

'background-repeat'

<i>Valor:</i>	repeat repeat-x repeat-y no-repeat <u>inherit</u>
<i>Inicial:</i>	Repeat
<i>Se aplica a:</i>	todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medios:</i>	<u>visual</u>
<i>Valor computado:</i>	Según lo especificado

Si se especifica una imagen de fondo, esta propiedad especifica si la imagen es repetida (mosaico), y de que modo. Todo el mosaico cubre las áreas de [contenido](#), [relleno](#) y [borde](#) de una caja.

El mosaico y la colocación de la imagen de fondo en elementos en línea no se define en esta especificación. Un nivel futuro de CSS puede definir el mosaico y la colocación de la imagen de fondo en elementos en línea.

Los valores tienen los significados siguientes:

repeat

La imagen se repite horizontal y verticalmente.

repeat-x

La imagen se repite solo horizontalmente.

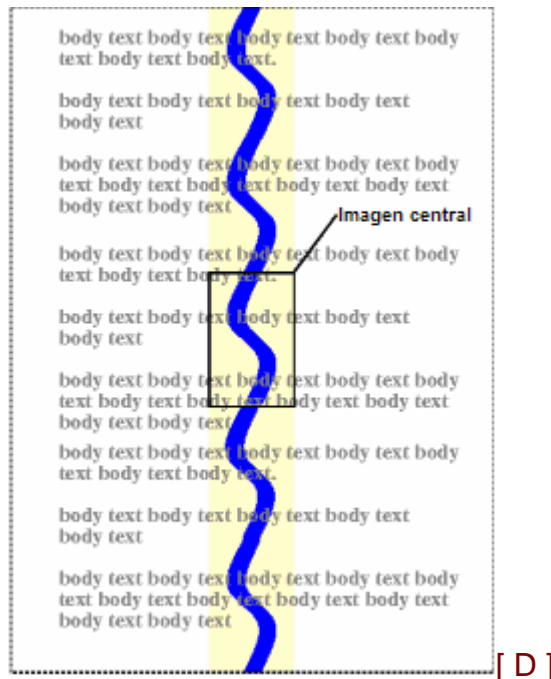
repeat-y

La imagen se repite solo verticalmente.

No-repeat

La imagen no se repite: solo una copia de la imagen se dibuja.

```
BODY {
  background: white url("pendiente.gif");
  background-repeat: repeat-y;
  background-position: center;
}
```



Una copia de la imagen de fondo se centra, y otras copias están colocadas por encima y por debajo de la misma para formar una banda vertical por detrás del elemento.

'background-attachment'

Valor:	scroll fixed <u>inherit</u>
Inicial:	Scroll
Se aplica a:	todos los elementos
Se hereda:	no
Porcentajes:	N/A
Medios:	<u>visual</u>
Valor computado:	Según lo especificado

Si se especifica una imagen de fondo, esta propiedad especifica si es fija con respecto al [acceso visual](#) ('fixed') o se desplaza junto con el bloque que contiene ('scroll').

Observe que hay solamente un acceso visual por opción. Si un elemento tiene un mecanismo de desplazamiento (véase 'desbordamiento'), el fondo 'fixed' no se mueve con el elemento, y un fondo 'scroll' no se mueve con el mecanismo de desplazamiento.

Incluso si la imagen es fija, sigue siendo visible solo cuando está en el fondo, en el área de relleno o en el borde del elemento. Así, a menos que la imagen forme un mosaico ('background-repeat: repeat'), puede ser invisible.

Este ejemplo crea una banda vertical infinita que permanece "pegada" al acceso visual cuando el elemento es desplazado.

```
BODY {
  background: red url("pendiente.gif");
  background-repeat: repeat-y;
  background-attachment: fixed;
}
```

Las aplicaciones del usuario que no soportan fondos 'fixed' (por ejemplo debido a las limitaciones de la plataforma del hardware) deben ignorar las declaraciones con la palabra clave 'fixed'. Por ejemplo:

```
body {
  background: white url(paper.png) scroll; /* Para todas las AUs */
  background: white url(ledger.png) fixed; /* Para AUs que soportan fondos
fijos */
}
```

Vea la sección en de [conformidad](#) para los detalles.

'background-position'

Valor: [[<porcentaje> | <medida> | left | center | right] [<porcentaje> | <medida> | top | center | bottom]?] || [[left | center | right] || [top | center | bottom]] | inherit

Inicial: 0% 0%

Se aplica a: Todos los elementos

Se hereda: no

Porcentajes: Referido al tamaño de la propia caja

Medio: visual

Valor computado: para <medida> el valor absoluto, sino un porcentaje

Si se ha especificado una imagen de fondo, esta propiedad especifica su posición inicial. Los valores tienen los significados siguientes:

< porcentaje > < porcentaje >

Con un par de valores de '0% 0%', la esquina superior izquierda de la imagen se alinea con la esquina superior izquierda del [límite del relleno](#) de la caja. Un par de valores de '100% 100%' pone la esquina inferior derecha de la imagen en la esquina inferior derecha del área de relleno. Con un par de valores de '14% 84%', el punto ubicado a un 14% del costado y a un 84% hacia abajo en la imagen debe ser colocado en el punto 14% del costado y a un 84% hacia abajo en el área de relleno.

< medida > < medida >

Con un par de valores de '2cm 2cm', la esquina superior izquierda de la imagen se pone 2cm a la derecha y 2cm debajo de la esquina superior izquierda del área de relleno.

Top left y left top

Igual que '0% 0%'.

Top, top center y center top

Igual que '50% 0%'.

Right top y top right

Igual que '100% 0%'.

Left, left center y center left

Igual que '0% 50%'.

Center y center center

Igual que '50% 50%'.

Right, right center y center right

Igual que '100% 50%'.

Bottom left y left bottom

Igual que '0% 100%'.

Bottom, bottom center y center bottom

Igual que '50% 100%'.

Bottom right y right bottom

Igual que '100% 100%'.

Si se da solamente un valor de porcentaje o de medida, éste fija solo la posición horizontal, y la posición vertical será el 50%. Si se dan dos valores, la posición horizontal viene primero. Las combinaciones de los valores de la palabra clave, medida y porcentaje se permiten, (Ej., '50% 2cm' o 'center 2cm' o 'center 10%'). Para las combinaciones de valores de palabra clave y de palabra no clave, 'left' y 'right' solo pueden ser utilizados como el primer valor, y 'top' y 'bottom' solo pueden ser utilizados como segundo valor. Se permiten las posiciones negativas.

```
body { background: url("banner.jpeg") right top } /* 100% 0% */
body { background: url("banner.jpeg") top center } /* 50% 0% */
body { background: url("banner.jpeg") center } /* 50% 50% */
body { background: url("banner.jpeg") bottom } /* 50% 100% */
```

El mosaico y la posicionamiento de la imagen de fondo en el elemento en línea no está definido en esta especificación. Un nivel futuro de CSS puede definir el mosaico y el posicionamiento de la imagen de fondo en elementos en línea.

Si la imagen de fondo es fija dentro del acceso visual (véase la propiedad '[background-attachment](#)'), la imagen se ubica en relación al acceso visual en vez de al área de relleno del elemento. Por ejemplo,

```
body {
background-image: url("logo.png");
background-attachment: fixed;
background-position: 100% 100%;
background-repeat: no-repeat;
}
```

En el ejemplo de arriba, la (única) imagen se pone en la esquina inferior derecha del acceso visual.

'background'

Valor: [<'background-color'> || <'background-image'> || <'background-repeat'> || <'background-attachment'> || <'background-position'>]
inheri

Inicial: vea las propiedades individuales

Se aplica a: todos los elementos

Se hereda: no

Porcentajes: permitido en la 'background-position'

Medios: visual

Valor computado: vea las propiedades individuales

La propiedad `'background'` es una propiedad resumida para fijar las propiedades individuales del fondo (es decir, `'background-color'`, `'background-image'`, `'background-repeat'`, `'background-attachment'` y `'background-position'`) en el mismo lugar en la hoja de estilo.

Dada una declaración válida, la propiedad `'background'` primero fija todas las propiedades individuales del fondo con sus valores iniciales, luego asigna los valores explícitamente datos en la declaración.

En la primera regla del ejemplo siguiente, solo un valor para `'background-color'` se ha fijado y las otras propiedades individuales se fijan con su valor inicial. En la segunda regla, se han especificado todas las propiedades individuales.

```
BODY { background: red }
P { background: url("chess.png") gray 50% repeat fixed }
```

14.3 Corrección de gamma

Para información sobre ediciones gammas, consulte por favor la guía de gamma en la especificación PNG ([\[PNG10\]](#)).

Nota. En el cómputo de la corrección gamma, AUs visualizadas con un CRT pueden asumir un CRT ideal e ignorar cualquier efecto sobre la gamma aparente causado por el dithering. Eso significa que el tratamiento mínimo que necesita hacer en la plataforma actual es:

PC usando MS-Windows

ninguno

Unix usando X11

ninguno

Mac usando QuickDraw

aplicar la gamma 1.45 [\[ICC32\]](#) (Las aplicaciones ColorSync-savvy pueden simplemente pasar el perfil del sRGB ICC a ColorSync para realizar la corrección adecuada del color)

Sgi usando X

aplicar el valor de gamma de `/etc/config/system.glGammaVal` (siendo el valor predeterminado de 1,70; las aplicaciones corriendo en Irix 6.2 o superior pueden pasar simplemente el perfil sRGB ICC al sistema de administración de color)

NeXTStep que usa siguiente

aplicar la gamma 2.22

"Aplicar una gamma" significa que cada uno de los tres R, G y B deben ser convertidos a $R'=R^{\text{gamma}}$, $G'=G^{\text{gamma}}$, $B'=B^{\text{gamma}}$, antes de ser entregados al OS. Esto puede efectuarse rápidamente construyendo una paleta de 256 elementos una vez por cada llamada al navegador, así:

```
for i := 0 to 255 do
  raw := i / 255.0;
  corr := pow (raw, gamma);
  table[i] := trunc (0.5 + corr * 255.0)
end
```

lo que entonces evita cualquier necesidad de efectuar cálculos complicados por atributo de color, mucho menos por píxel.

15 Fuentes

Contenido

- [15.1 Introducción](#)
- [15.2 Algoritmo de equivalencia de fuentes](#)
- [15.3 Familias de fuentes: la propiedad 'font-family'](#)
- [15.4 Estilos de la fuente: la propiedad 'font-style'](#)
- [15.5 Small-caps: la propiedad 'font-variant'](#)
- [15.6 Grosor de la fuente: la propiedad 'font-weight'](#)
- [15.7 Tamaño de fuente: la propiedad 'font-size'](#)
- [15.8 Propiedad abreviada de fuente: la propiedad 'font'](#)

15.1 Introducción

Fijar propiedades de la fuente estará entre las aplicaciones más comunes de las hojas de estilo. Desafortunadamente, no existe ninguna taxonomía bien definida y universalmente aceptada para clasificar fuentes, y los términos que se aplican a una familia de la fuente pueden no ser apropiados para otras. Por ejemplo 'itálic' se utiliza comúnmente para etiquetar el texto inclinado (cursiva), pero el texto inclinado se puede también etiquetar como *Oblique (oblicuo)*, *Slanted (inclinado)*, *Incline (pendiente)*, *Cursive (cursivo)* o *Kursiv*. Por lo tanto no es un simple problema de trazar un mapa de propiedades de fuentes típicas de selección de una fuente específica.

15.2 Algoritmo de equivalencia de fuentes

Porque no hay aceptado, una taxonomía universal de las propiedades de la fuente, la equivalencia de las propiedades del aspecto de la fuente se debe hacer cuidadosamente. Las propiedades son equivalentes en un orden bien definido para asegurar que los resultados de este proceso de equivalencia es tan constante como sea posible a través de las AUs (si se asume que la misma biblioteca del aspecto de la fuente está presentada a cada uno de ellas).

1. La aplicación del usuario hace (o accede) a una base de datos de las propiedades relevantes de CSS 2.1 de todas las fuentes de las cuales el AU reconoce. Si hay dos fuentes exactamente con las mismas propiedades, la aplicación del usuario selecciona una de ellas.
2. En un elemento dado y para cada carácter en ese elemento, la AU integra las propiedades de la fuente aplicables a ese elemento. Usando el sistema completo de propiedades, la AU utiliza la propiedad 'font-family' para elegir una familia de fuentes tentativa. Las propiedades restantes se prueban con la familia según los criterios de equivalencia descritos en cada propiedad. Si hay equivalencia para todas las propiedades restantes, entonces ése es el aspecto de la fuente equivalente para el elemento dado.
3. Si no hay ninguna equivalente para el aspecto de la fuente dentro de 'font-family' procesada en el paso 2, y si hay una 'font-family' siguiente alternativa en la fuente fijada, entonces se repite el paso 2 con la siguiente 'font-family' alternativa.
4. Si hay equivalencia para la fuente, pero no contiene un glyph para el carácter actual, y si hay una 'font-family' siguiente alternativa a la fuente fijada, entonces se repite el paso 2 con la 'font-family' alternativa siguiente.

5. Si no hay ninguna fuente dentro de la familia seleccionada en 2, entonces utiliza una 'font-family' por defecto dependiente de la AU y se repite el paso 2, utilizando la mejor equivalencia que pueda ser obtenida dentro de la fuente por defecto. Si un carácter en particular no se puede mostrar utilizando esta fuente, entonces la AU no tiene ninguna fuente apropiada para ese carácter. La AU debe mapear cada carácter para el cual no tiene ninguna fuente conveniente a un símbolo visible elegido por el AU, preferiblemente un glyph del "carácter que falta" a partir de una de las apariencias de la fuente disponible en la AU.

(El algoritmo anterior se puede optimizar para evitar tener que visitar las propiedades del CSS 2.1 para cada carácter.)

Las reglas de equivalencia por propiedad a partir de (2) anterior son las siguientes:

1. 'font-style' se intenta primero. 'italic' será satisfecha si hay fuente en la base de datos de fuentes del AU definida con la palabra clave de CSS 'italic' (preferida) o 'oblique'. Si no los valores deben ser exactamente equivalentes o font-style fallará.
2. 'font-variant' se intenta después. 'small-caps' (1) equivale a una fuente designada como 'small-caps', (2) una fuente en la cual las mayúsculas pequeñas son sintetizadas, o (3) una fuente donde todas las letras minúsculas son reemplazadas por las letras mayúsculas. Una fuente small-caps puede ser sintetizada electrónicamente escalando letras mayúsculas de una fuente normal. 'Normal' equivale a una variante normal de fuente(no-small-caps). Una fuente no puede tener una variante normal. Una fuente que está solo disponibles como small-caps será selecciona como fuente 'normal' o 'small-caps'.
3. 'font-weight' es equivalente después, nunca será fallido. (véase 'font-weight' abajo.)
4. 'font-size' debe ser equivalente dentro de un margen de tolerancia dependiendo de la AU. (típicamente, los tamaños para las fuentes escalables se redondean al píxel entero más cercano, mientras que la tolerancia para las fuentes mapas de bits podrían ser tan grande como el 20%.) Otros cálculos, ej. para valores 'em' en otras propiedades, son basados en el valor computado de 'font-size'.

15.3 Familia de la fuente: la propiedad 'font-family'

'font-family'

<i>Valor:</i>	<code>[[<nombre de la familia> <familia genérica>] [, <nombre de la familia> <familia genérica>]*] inherit</code>
<i>Inicial:</i>	Depende de la aplicación del usuario
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Como el especificado

El valor es una lista de prioridades de nombres de fuentes de familias de fuentes y/o nombres de familia genéricas. A diferencia de la mayoría de las otras propiedades de CSS, los valores son separados por una coma para indicar que son alternativas:

```
body { font-family: Gill, Helvetica, sans-serif }
```

Aunque muchas fuentes proporcionan el glyph del "carácter que falta", típicamente como una caja abierta, como su nombre implica esto no se debe considerar como una equivalencia para los caracteres que no se pueden encontrar en la fuente. (debe, sin embargo, ser considerada una equivalencia para U+FFFD, el punto de código del "carácter que falta").

Hay dos tipos de nombres de familias de fuentes:

<nombre de la familia>

El nombre de una familia de fuentes de preferencia. En el ejemplo anterior, "Gill" y "helvetica" son familias de fuentes.

<familia genérica>

En el ejemplo de arriba, el último valor es un nombre de familia genérico. Los siguientes nombres de familias genéricas se definen:

- ' serif ' (ej. Times)
- ' sans-serif ' (ej. helvetica)
- ' cursivo ' (ej. Zapf-Chancery)
- ' fantasía ' (ej. Western)
- ' monospace ' (Ej. Courier)

Se anima a los diseñadores de la hoja de estilo a que ofrezcan una familia genérica de fuente como última alternativa. Los nombres de familia genéricos de fuentes son palabras claves y no deben ir entre comillas.

Si un nombre de familia que no está entre comillas contiene paréntesis, corchetes y/o llaves, deben ser balanceados o escapados por reglas de la gramática de CSS. Igualmente, las comillas, punto y comas, signos de exclamación y comas dentro de los nombres de familia de fuentes que no están entre comillas deben ser escapados. Los nombres de fuentes que contienen cualesquiera otro carácter o espacio en blanco deben ir entre comillas.

```
body { font-family: "New Century Schoolbook", serif }
```

```
<BODY STYLE="font-family: 'My own font', fantasy">
```

Si se omite las comillas, cualquier carácter de espacio en blanco antes y después el nombre de la fuente es ignorado y cualquier secuencia de los caracteres de espacio en blanco dentro del nombre de la fuente se convierte a un solo espacio. Los nombres de familias *de fuentes* que pueden ser iguales que un valor de una palabra clave (ej. 'initial', 'inherit', 'default', 'serif', 'sans-serif', 'monospace', 'fantasy', y 'cursive') se deben poner entre comillas para prevenir la confusión con las palabras claves. Las AUs no deben considerar estas palabras claves como equivalentes al tipo de '<nombre de familia>' tipo.

15.4 Estilos de la fuente: la propiedad 'font-style'

'font-style'

Valor:	normal italic oblique <u>inherit</u>
Inicial:	normal
Se aplica a:	todos los elementos
Se hereda:	sí
Porcentajes:	N/A
Medios:	<u>visual</u>
Valor computado:	según lo especificado

La propiedad 'font-style' selecciona entre el aspecto normal (designado a veces "roman" o "upright"), Italia (italica) y oblique (oblicua) dentro de una familia de fuentes.

Un valor 'normal' selecciona una fuente que se clasifique como 'normal' en la base de datos de fuentes de la AU, mientras que 'oblique' selecciona una fuente que se clasifica como 'oblique'. Un valor del 'italic' selecciona una fuente que se clasifica como 'italic', o, si ésta no está disponible, una definida como 'oblique'.

La fuente que definida como 'oblique' en la base de datos de fuentes de la AU puede haber sido generada en realidad inclinando electrónicamente una fuente normal.

Las fuentes con oblique, slanted o incline en sus nombres será típicamente definidas como 'oblique' en la base de datos de fuentes de la AU. Las fuentes con *italic*, *cursive* o *Kursiv* en sus nombres será típicamente definidas como 'italic'.

```
h1, h2, h3 { font-style: italic }
h1 em { font-style: normal }
```

En el ejemplo de arriba, el texto enfatizado dentro de 'H1' aparecerá en un aspecto normal.

15.5 Small-caps: la propiedad 'font-variant'

'font-variant'

Valor:	normal small-caps <u>inherit</u>
Inicial:	normal
Se aplica a:	todos los elementos
Hereditado:	sí
Porcentajes:	N/A
Medios:	<u>visual</u>
Valor computado:	según lo especificado

Otro tipo de variación dentro de una familia de fuentes es el small-caps (versales o versalitas). En una fuente small-caps las letras minúsculas parecen similares las mayúsculas, pero en un tamaño más pequeño y con proporciones ligeramente diferentes. La propiedad 'font-variant' selecciona esa fuente.

Un valor 'normal' selecciona una fuente que no está definida como una fuente small-caps (versalita), 'small-caps' selecciona una fuente de versalita (small-caps). Es aceptable (pero no requerido) en CSS 2.1 que la fuente small-caps (versalita) sea creada tomando una fuente normal y reemplazando las letras minúsculas por los caracteres mayúsculas reducidos. Como último recurso, las letras mayúsculas serán utilizadas como reemplazo para una fuente versalita (small-caps).

El ejemplo siguiente da lugar a un elemento 'H3' con small-caps (versalita), con cualquier palabra enfatizada en palabras oblicuas (oblique), y cualquier palabra enfatizada dentro de un 'H3' versalita (small-caps) oblicuo (oblique):

```
h3 { font-variant: small-caps }
em { font-style: oblique }
```

Puede haber también otras variantes en la familia de fuentes, tales como fuentes con letras numerales del viejo-estilo (old-style), numerales versalitas (small-caps), condensadas o ampliadas, etc. CSS 2.1 no tiene ninguna propiedad que seleccione ésas.

Nota: En la medida en que esta propiedad provoca que el texto sea transformado a mayúscula, se aplican las mismas consideraciones que para 'text-transform'.

15.6 Grosor de la fuente: la propiedad 'font-weight'

'font-weight'

Valor: normal|bold|bolder|lighter|100|200|300|400|500|600|700|800|900|inherit
Inicial: normal
Se aplica a: todos los elementos
Se Hereda: sí
Porcentajes: N/A
Medios: visual
Valor computado: vea el texto

La propiedad 'font-weight' selecciona el peso de la fuente. Los valores entre '100' a '900' forman una secuencia ordenada, donde cada número indica un peso que es al menos tan oscuro (gruesa) como su predecesor. La palabra clave '**normal**' es igual que '400', y '**bold**' es igual que '700'. Las palabras claves con excepción de 'normal' y 'bold' que se han mostrado a menudo son confundidas con nombres de fuentes y por lo tanto una escala numérica fue elegida para los 9 valores de la lista.

```
p { font-weight: normal } /* 400 */
h1 { font-weight: 700 } /* bold */
```

Los valores 'bolder' y 'lighter' seleccionan los pesos de fuentes que están relacionados al peso heredado del padre:

```
strong { font-weight: bolder }
```

Los elementos hijos heredan el peso resultante, no el valor de la palabra clave.

Las fuentes (los datos de la fuente) tienen típicamente una o más propiedades cuyos valores son nombres descriptivos de "weight" (peso) de una fuente. No hay un significado universalmente aceptado para estos nombres del peso. Su papel principal es distinguir aspectos de diferente oscuridad (grosor) dentro de una sola familia de fuentes. El uso a través de las familias de fuentes es absolutamente variable; por ejemplo, una fuente en la cual uno puede pensar como sería el grosor podría ser descrita como *Regular, Roman, Book, Medium, Semi- o DemiBold, Bold, or Black, dependiendo de cómo de negra es el aspecto de una fuente designada con "normal"*. Porque no hay uso estándar de nombres, los valores de propiedad del peso en CSS 2.1 se dan en una escala numérica en la cual el valor '400' (o 'normal') corresponda al aspecto "normal" del texto para esa familia. El nombre del peso se asocia a ese aspecto será típicamente *Book, Regular, Roman, Normal* o a veces *Medium*.

La asociación de otros pesos dentro de una familia a los valores de peso numéricos se piensa solo para preservar orden de oscuridad (grosor) dentro de aquella familia. Sin embargo, la heurística siguiente dice cómo la asignación se hace en casos típicos:

- Si la familia de fuentes ya utiliza una escala numérica con nueve valores (como por ej. OpenType), los pesos de la fuente deberán ser mapeados directamente.
- Si hay tanto un aspecto designado como *Medium* y uno como *Book, Regular, Roman* o *Normal*, entonces *Medium* es normalmente asignado a '500'.
- La fuente designado con "Bold" corresponderá al valor del peso '700'.
- Si hay menos de 9 pesos en la familia, el algoritmo por defecto para llenar los "agujeros" (los que faltan hasta completar los 9) es el siguiente. Si '500' no está asignado, se asignara la misma fuente como '400'. Si algunos de los valores '600', '700', '800' o '900' están sin asignar, estos serán asignados al mismo aspecto que la siguiente palabra clave asignada, si la hay, o el siguiente mas claro si no. Si cualquiera de '300', '200' o '100' no es asignado, se asignara la palabra clave siguiente mas clara, si la hay, o la siguiente mas oscura si no.

Los siguientes dos ejemplos muestran un mapeado típico.

Asuma cuatro pesos en la familia "Rattlesnake", desde mas claro a más oscuro: *Regular, Medium, Bold, Heavy*.

<i>Primer ejemplo de un mapeado del peso de la fuente</i>		
Posibles aspectos	Asignaciones	Relleno de los agujeros
"Rattlesnake Regular"	400	100, 200, 300
"Rattlesnake Medium"	500	
"Rattlesnake Bold"	700	600
"Rattlesnake Heavy"	800	900

Asuma seis peso en la familia "Ice Prawn": *Book, Medium, Bold, Heavy, Black, ExtraBlack*. Observe que en este caso la aplicación del usuario no tiene que asignar un valor numérico a "Ice Prawn ExtraBlack".

Segundo ejemplo de mapeado de peso de fuente

Posibles aspectos	Asignaciones	Relleno de agujeros
"Ice Prawn Book"	400	100, 200, 300
"Ice Prawn Medium"	500	
"Ice Prawn Bold"	700	600
"Ice Prawn Heavy"	800	
"Ice Prawn Black"	900	
"Ice Prawn ExtraBlack"	(none)	

Ya que la intención de las palabras claves relativas 'bolder' y 'lighter' es la de oscurecer o aclarar el aspecto dentro de la familia y porque una familia no puede tener aspectos alineados con todos los valores de pesos simbólicos, la equivalencia de 'bolder' es el siguiente aspecto oscuro disponible dentro de la familia en el cliente y la equivalencia de 'lighter' es el siguiente aspecto mas claro en la familia. Para ser exactos, el significado de las palabras claves relativas 'bolder' y 'lighter' son los siguientes:

- **'bolder'** Selecciona el siguiente peso que es asignado a una fuente que es mas oscura que el heredado. Si no existe tal peso, el resultado es simplemente el siguiente valor numérico más oscuro (y la fuente permanece sin cambios), a menos que el valor heredado fuese '900' en cuyo caso el peso resultante es también '900'.
- **'lighter'** es similar, pero trabaja en la dirección contraria: Selecciona la siguiente palabra clave mas clara con una fuente diferente del heredado, a menos que no haya tal fuente, en tal caso selecciona el siguiente valor numérico más claro (y la fuente se mantiene sin cambios).

No hay garantía de que habrá un aspecto mas oscuro para un valor de 'font-weight'; por ejemplo, algunas fuentes pueden tener solo un aspecto normal y fuerte (bold), mientras que otras pueden tener ocho pesos de aspectos. No hay granita sobre como una AU puede trazar un mapa (mapear) del aspecto de la fuente dentro de una familia a los valores de peso. La única garantía es que el aspecto de un valor dado no será menos oscura que el aspecto de un valor claro.

El valor computado de "font-weight" es:

- Uno de los valores de número legales, o
- Uno de las combinaciones de valores de numero legales con uno o mas de los valores relativos (bolder o lighter). Este tipo de valores computado es necesario utilizarlo cuando la fuente en cuestión no tiene todas las variantes de peso que son necesarias.

CSS 2.1 no especifica como el valor computado de font-weight es representado interna o externamente.

15.7 Tamaño de la fuente: la propiedad 'font-size'

'font-size'

Valor:	<u><tamaño-absoluto></u> <u><tamaño-relativo></u> <u><medida></u> <u><porcentaje></u> <u>inherit</u>
Inicial:	medium
Se aplica a:	Todos los elementos
Se hereda:	Si
Porcentajes:	Referido al tamaño de la fuente del elemento padre
Medio:	<u>visual</u>
Valor computado:	Medida absoluta

El tamaño de la fuente se corresponde con la escala 'em', un concepto utilizado en tipografía. Observe que ciertos glyphs pueden ser sangrados fuera de la escala 'em'. Los valores tienen los siguientes significados:

<tamaño-absoluto>

Una palabra clave <tamaño-absoluto> es un índice en una tabla de tamaños de fuentes computadas y guardados por la AU. Los posibles valores son:

[xx-small | x-small | small | medium | large | x-large | xx-large]

La siguiente tabla proporciona unas directivas a la aplicación del usuario para el tamaño-absoluto escalado por el factor y su mapeado a las etiquetas HTML y los tamaños de fuentes absolutos. El valor 'medium' es utilizado como referencia del valor medio de referencia. La aplicación del usuario puede afinar estos valores para diferentes fuentes o diferentes tipos de dispositivos que la muestran.

CSS valor de tamaño absoluto	xx-small	x-small	small	medium	large	x-large	xx-large	
Factor de escala	3/5	3/4	8/9	1	6/5	3/2	2/1	3/1
Etiqueta HTML	h6		h5	h4	h3	h2	h1	
Tamaños de fuente HTML	1		2	3	4	5	6	7

Los diferentes medios pueden necesitar diferentes factores de escala. Además, la AU debe tener en cuenta la calidad y disponibilidad de las fuentes consideradas cuando la tabla es computada. La tabla puede ser diferente de una familia a otra.

Nota 1. Para mantener la legibilidad, una AU aplicando estas directivas sin embargo debería evitar crear tamaños de fuentes menores a 9 píxeles por unidad EM sobre el resultado mostrado en el ordenador..

Nota 2. En CSS1, el factor de escala sugerido entre los índices adyacentes era 1.5 que la experiencia de los usuarios demostró que era demasiado grande. En CSS2 el factor de escala sugerido para la pantalla del ordenador entre índices adyacentes fue de 1.2 que aun crea dudas para los tamaños pequeños. El factor de escala varía entre cada índice para proporcionar la mejor legibilidad.

<tamaño-relativo>

Una palabra clave <tamaño-relativo> es interpretada como relativa a la tabla de tamaños de fuente y al tamaño de la fuente del elemento padre. Los posibles valores son: [**larger** | **smaller**]. Por ejemplo, si el elemento padre tiene un tamaño de fuente 'medium', un valor 'larger' hará que el tamaño de la fuente del elemento actual sea 'large'. Si el tamaño del elemento padre no es cercano a una entrada de la tabla, la AU es libre de interpolarlo entre las entradas de la tabla o redondearlo al más cercano. La AU puede tener que extrapolar valores de la tabla si los valores numéricos sobrepasan las palabras claves.

Los valores <medida> y <porcentaje> no deben tener en cuenta la tabla de tamaños de fuente en el cálculo cuando calculan el tamaño de la fuente de un elemento.

Los valores negativos no son permitidos.

Sobre el resto de propiedades, las medidas de los valores 'em' y 'ex' se refieren al tamaño de fuente computado del elemento actual. Sobre la propiedad 'font-size', estas unidades de medida se refieren al tamaño de fuente computado del elemento padre.

Observe que una aplicación puede reinterpretar un tamaño explícito, dependiendo del contexto. Por ejemplo, dentro de una escena VR una fuente puede tomar un tamaño diferente por la distorsión de la perspectiva.

Examples:

```
p { font-size: 16px; }
@media print {
  p { font-size: 12pt; }
}
blockquote { font-size: larger }
em { font-size: 150% }
em { font-size: 1.5em }
```

15.8 Propiedad abreviada de la fuente: La propiedad **'font'**

'font'

<i>Valor:</i>	[[<font-style> <font-variant> <font-weight>]? <font-size> [/ <line-height>]? <font-family>] caption icon menu message-box small-caption status-bar inherit
<i>Inicial:</i>	Vea las propiedades individuales
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	Ver propiedades individuales
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Ver propiedades individuales

La propiedad 'font' es, exceptuando los casos descritos [abajo](#), un resumen de la propiedad para determinar 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' y 'font-family' en un mismo lugar en la hoja de estilo. La sintaxis de esta propiedad está basada en una tipografía tradicional de notación resumida para poner múltiples propiedades relacionadas con la fuente.

Todas las propiedades relacionadas con las fuentes son primero reseteadas a sus valores iniciales, incluyendo aquellas listadas en el párrafo anterior. Entonces, aquellas propiedades que tienen asignados valores explícitos en la forma resumida `'font'` son establecidas con esos valores. Para una definición de los valores permitidos e iniciales, vea las propiedades definidas anteriormente:

```
p { font: 12px/14px sans-serif }
p { font: 80% sans-serif }
p { font: x-large/110% "New Century Schoolbook", serif }
p { font: bold italic large Palatino, serif }
p { font: normal small-caps 120%/120% fantasy }
```

En la segunda regla, el valor del porcentaje del tamaño de la fuente ('80%') se refiere al tamaño de la fuente del elemento padre. En la tercera regla, el porcentaje de la altura de la línea se refiere al tamaño de fuente del propio elemento

En las tres primeras reglas de arriba, `'font-style'`, `'font-variant'` y `'font-weight'` no son mencionadas explícitamente, lo que significa que las tres tomarán sus valores iniciales ('normal'). La cuarta regla fija el `'font-weight'` a `'bold'`, el `'font-style'` a `'italic'` e implícitamente fija `'font-variant'` a `'normal'`.

La quinta regla establece `'font-variant'` ('small-caps'), `'font-size'` (120% de la fuente del padre), `'line-height'` (120% veces el tamaño de la fuente) y `'font-family'` ('fantasy'). Se entiende que la palabra clave `'normal'` se aplica a las dos propiedades restantes: `'font-style'` y `'font-weight'`.

Los siguientes valores se refieren a las fuentes del sistema:

caption

La fuente utilizada para los títulos de los controles (ej., botones, desplegados verticales (drop-downs), etc.).

icon

La fuente utilizada para rotular los iconos.

menu

La fuente utilizada en los menús (menús desplegados verticalmente y listas de menús).

message-box

La fuente utilizada por las cajas de diálogo.

small-caption

La fuente utilizada para rotular controles pequeños.

status-bar

La fuente utilizada en la barra de estado de las ventanas.

Las fuentes del sistema solo pueden asignarse como un conjunto, es decir, la familia, el tamaño, el peso, el estilo, etc son establecidos al mismo tiempo. Estos valores pueden entonces modificarse individualmente si se desea. Si en determinada plataforma no existe una fuente con las características indicadas, la aplicación del usuario debe realizar una sustitución inteligente (ej. Una versión mas pequeña de la fuente para `'caption'` puede ser usada como fuente para `'small-caption'`), o sustituir la fuente predeterminada de la aplicación del usuario. Al igual que para las fuentes regulares, si, para una fuente del sistema, cualquiera de las propiedades individuales no forman parte de las preferencias del usuario disponibles en el sistema operativo, esas propiedades deberán ser fijadas en sus valores iniciales.

Eso es porque esta propiedad es "casi" una propiedad resumida: las fuentes del sistema solo pueden ser especificadas con esta propiedad, no con 'font-family' en si, de manera que 'font' permite a los autores hacer más cosas que con la suma de sus subpropiedades. Sin embargo, las propiedades individuales como 'font-weight' aun están dando valores tomados de la fuente del sistema, los cuales pueden ser variados independientemente.

```
button { font: 300 italic 1.3em/1.7em "FB Armada", sans-serif }
button p { font: menu }
button p em { font-weight: bolder }
```

Si la fuente usada por los menús desplegables verticalmente en un sistema en particular pasó a ser, por ejemplo, la Charcoal de 9 puntos, con un peso de 600, entonces los elementos P que sean descendentes de BUTTON serían mostrados como si esta regla estuviera en efecto:

```
button p { font: 600 9px Charcoal }
```

Porque la propiedad resumida 'font' resetea cualquier propiedad sin un valor explicito dado al valor inicial, esta tiene el mismo efecto que esta declaración:

```
button p {
  font-family: Charcoal;
  font-style: normal;
  font-variant: normal;
  font-weight: 600;
  font-size: 9px;
  line-height: normal;
}
```

16 Texto

Contenido

- [16.1 Sangría: la propiedad 'text-indent'](#)
- [16.2 Alineación: la propiedad 'text-align'](#)
- [16.3 Decoración](#)
 - [16.3.1 Subrayado, sobrerayado, tachado, y parpadeo: la propiedad 'text-decoration'](#)
- [16.4 Espaciado de letras y palabras: las propiedades 'letter-spacing' y 'word-spacing'](#)
- [16.5 Capitalización: la propiedad 'text-transform'](#)
- [16.6 Espacios en blanco: la propiedad 'white-space'](#)
 - [16.6.1 Modelo de procesado de 'white-space'](#)
 - [16.6.2 Ejemplo de bidireccionalidad con white-space cerrados](#)

Las propiedades definidas en las secciones siguientes afectan a la presentación visual de caracteres, de espacios, de palabras, y de párrafos.

16.1 Sangría: La propiedad 'text-indent'

'text-indent'

<i>Valor:</i>	<code><medida> <porcentaje> inherit</code>
<i>Inicial:</i>	0
<i>Se aplica a:</i>	Elementos a nivel de bloque, celdas de tabla y bloques en línea
<i>Se hereda:</i>	sí
<i>Porcentajes:</i>	refiero a la anchura de bloque de contención
<i>Medios:</i>	<u>visual</u>
<i>Valor computado:</i>	El porcentaje según lo especificado o la medida absoluta

Esta propiedad especifica la sangría de la primera línea del texto en un bloque. Más exacto, especifica la sangría de la primera caja que fluye dentro de la primera [caja de línea](#) dentro del bloque. La caja está sangrada con respecto al límite izquierdo (o derecho, para las composiciones de derecha a izquierda) de la caja de línea. Las aplicaciones del usuario deben procesar este sangrado como espacio en blanco.

Los valores tienen los significados siguientes:

<medida>

El sangrado es una medida fija.

<porcentaje>

El sangrado es un porcentaje del ancho del bloque de contención.

El valor de `'text-indent'` puede ser negativo, pero puede haber limitaciones específicas de la implementación. Si el valor de `'text-indent'` es negativo o excede la anchura del bloque, esa *primera caja*, descrita arriba, puede desbordar el bloque. El valor de `'overflow'` afectará si tal texto que desborda el bloque es visible.

El ejemplo siguiente causa una sangría de '3em' en el texto.

```
p { text-indent: 3em }
```

Nota: Puesto que la propiedad 'text-indent' se hereda, cuando está especificada en un elemento de bloque, afectará a los elementos de bloque en línea descendientes. Por esta razón, es a menudo especificado 'text-indent: 0' en los elementos que son especificados 'display:inline-block'.

16.2 Alineación: la propiedad 'text-align'

'text-align'

Valor:	left right center justify <u>inherit</u>
Inicial:	'left' si 'direction' es 'ltr'; 'right' si 'direction' es 'rtl'
Se aplica a:	Elementos a nivel de bloque, celdas de tabla y bloques en línea
Se hereda:	Si
Porcentajes:	N/A
Medio:	<u>visual</u>
Valor computado:	Según el especificado

Esta propiedad describe cómo el contenido en línea de un bloque se alinea. Los valores tienen los siguientes significados:

Left, right, center, justify

Texto alineado a la izquierda, derecha, centro y justificado, respectivamente

Un bloque de texto es una pila de [cajas de línea](#). En el caso de 'left', 'right' y 'center', esta propiedad especifica cómo las cajas en línea dentro de cada caja de línea se alinean con respecto a los lados izquierdo y derecho de la caja de línea; la alineación no está con respecto al [acceso visual](#). En el caso de 'justify', la AU puede estirar las cajas en línea además de ajustar sus posiciones. (véase también '[letter-spacing](#)' y '[word-spacing](#)').)

Si es el valor computado de text-align es 'justify' mientras que el valor computado de white-space (espacio en blanco) es 'pre' o 'pre-line', el valor real de text-align (alineación del texto) se fija al valor inicial.

En este ejemplo, observe que mientras 'text-align' es heredado, todos los elementos a nivel de bloque dentro del elemento DIV con 'class=important' tendrá su contenido en línea centrado.

```
div.important { text-align: center }
```

Nota. El algoritmo real de justificación usado depende de la aplicación del usuario y del language/script del texto.

[Las aplicaciones del usuario con conformidad](#) pueden interpretar el valor 'justify' como 'left' o 'right', dependiendo si la dirección de la escritura predeterminada del elemento es de izquierda a derecha o derecha a izquierda, respectivamente.

16.3 Decoración

16.3.1 Subrayado, sobrrayado, tachado, y parpadeo: la propiedad **'text-decoration'**

'text-decoration'

<i>Valor:</i>	none [underline overline line-through blink] <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no (ver texto)
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Según el especificado

Esta propiedad describe las decoraciones que se agregan al texto de un elemento usando el color del elemento. Cuando está especificado en un elemento en línea, afecta a todas las cajas generadas por ese elemento; para el resto de los elementos, las decoraciones se propagan a una caja en línea anónima que envuelva a todos los hijos en línea en el flujo del elemento, y a cualquier descendente [nivel de bloque](#) en flujo. No es, sin embargo, propagado más allá de los descendentes posicionados como flotantes y absolutos, ni al contenido de descendentes 'inline-table' (tablas en línea) y 'inline-block' (bloques en línea).

Si un elemento no contiene ningún texto (ignorando los espacios en blancos en elementos que tienen 'white-space' fijado a 'normal', 'pre-line', o 'no-wrap'), las aplicaciones del usuario deben frenarse de procesar las decoraciones del texto en el elemento. Por ejemplo, los elementos que contienen solamente imágenes y espacios blanco cerrados no serán subrayados.

Las decoraciones del texto en las cajas en línea se dibujan a través del elemento entero, yendo a través de cualquier elemento descendiente sin prestar ninguna atención a su presencia. La propiedad 'text-decoration' en elementos descendiente no puede tener ningún efecto en la decoración del elemento. En la determinación de la posición y del grosor de las líneas de la decoración del texto, las aplicaciones del usuario pueden considerar los tamaños de la fuente y las líneas del fondo dominantes de descendientes, pero deben utilizar la misma línea de fondo y grosor en cada línea.

Los valores tienen los significados siguientes:

none

No produce ninguna decoración en el texto.

underline

Cada línea del texto se subraya.

overline

Cada línea del texto tiene una línea sobre ella.

Line-through

Cada línea del texto tiene una línea por el centro (tachándola).

parpadeo

El texto parpadea (alterna entre visible e invisible). [Las aplicaciones del usuario con conformidad](#) no están obligadas a soportar este valor. Observe que el parpadeo del texto es una técnica para satisfacer [el punto de comprobación 3.3 de Wai-uaag](#).

El(los) color(es) requerido para la decoración del texto se deben derivar del valor de la propiedad 'color' del elemento en el cual se fija 'text-decoration'. El color de decoraciones debe seguir siendo el mismo (uniforme) si los elementos descendientes tienen diversos valores de 'color'.

Algunas aplicaciones del usuario han implementado text-decoration propagando la decoración a los elementos descendientes a diferencia de simplemente el dibujo de la decoración a través de los elementos según lo descrito arriba. Esto fue permitido por la expresión más floja en CSS2. Las aplicaciones del usuario SVG1, CSS1-solo, y CSS2-solo pueden poner el modelo más viejo en practica y todavía demandar conformidad a esta parte de CSS2.1. (esto no se aplica a las AUs desarrolladas después de que esta especificación fuera lanzada.)

En el siguiente ejemplo para HTML, el contenido del texto de todos los elementos A que actúan como hipervínculos (si está visitado o no) serán subrayado:

```
a:visited,a:link { text-decoration: underline }
```

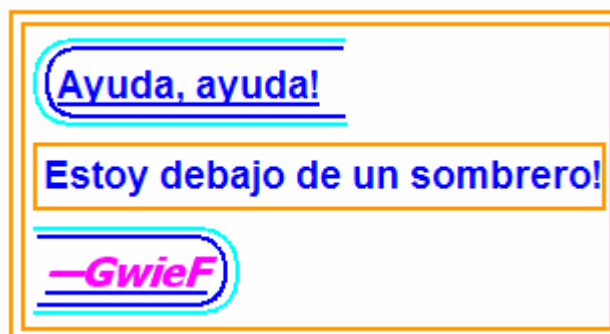
En la siguiente de hoja de estilo y fragmento de documento:

```
blockquote { text-decoration: underline; color: blue; }
em { display: block; }
cite { color: fuchsia; }

<blockquote>
  <p>
    <span>
      Ayuda, ayuda!
      <em> Estoy debajo de un sombrero! </em>
      <cite> -GwieF </cite>
    </span>
  </p>
</blockquote>
```

... el subrayado para el elemento blockquote se propaga a un elemento en línea anónimo que rodea al elemento span, causando que el texto "ayuda, ayuda!" sea azul, con subrayado azul del anónimo en línea por debajo de él, el color que es tomado del elemento blockquote.

texto en el bloque em no se subrayan ninguno, pues que no está contenido en el mismo elemento en línea anónimo. La línea final del texto es fucsia, pero el subrayado de esta sigue siendo una raya azul del elemento en línea anónimo.



Este diagrama muestra las cajas implicadas en el ejemplo de arriba. La línea redondeada de **agua** representa el elemento en línea anónimo que envuelve el contenido en línea del elemento del párrafo, la línea **azul** redondeada representa el elemento span, y las líneas **anaranjadas** representan los bloques.

16.4 Espaciado de letras y de palabras: la propiedad **'letter-spacing'** y **'word-spacing'**

'letter-spacing'

<i>Valor:</i>	normal <u><medida></u> <u>inherit</u>
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	'normal' o medida absoluta

Esta propiedad especifica el comportamiento de los espacios entre los caracteres del texto. Los valores tienen los siguientes significados:

normal

El espaciado es el espacio normal para la fuente actual. Este valor permite que la aplicación del usuario altere el espacio entre los caracteres para justificar el texto.

<medida>

Este valor indica el espacio entre caracteres *además* del espacio por defecto entre los caracteres. Los valores pueden ser negativos, pero puede haber límites específicos de la implementación. Las aplicaciones del usuario no puede aumentar o disminuir el espacio entre caracteres para justificar el texto.

Los algoritmos del espaciado de caracteres dependen de las aplicaciones del usuario.

En este ejemplo, el espacio entre los caracteres en los elementos **BLOCKQUOTE** es aumentado en ' 0.1em '.

```
blockquote { letter-spacing: 0.1em }
```

En el ejemplo siguiente, no se le permite a la aplicación del usuario alterar el espacio entre caracteres:

```
blockquote { letter-spacing: 0cm }/* iguales a '0' */
```

Cuando el espacio resultante entre dos caracteres no es igual que el espacio por defecto (predeterminado), las aplicaciones del usuario no deben utilizar ligaduras.

'word-spacing'

<i>Valor:</i>	normal <u><medida></u> <u>inherit</u>
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	para 'normal' el valor '0'; si no la medida absoluta

Esta propiedad especifica el comportamiento del espacio entre las palabras. Los valores tienen los significados siguientes:

normal

El espacio normal entre palabras, según lo definido por la fuente actual y/o la AU.

<medida>

Este valor indica el espacio entre palabras *además* del espacio entre las palabras por defecto. Los valores pueden ser negativos, pero puede haber límites específicos puestos por la implementación.

Los algoritmos del espaciado entre palabras dependen de la aplicación del usuario. El espaciado de la palabra también está influenciado por la justificación (véase la propiedad 'text-align'). Cuando se preserva el espacio blanco, Ej. con 'white-space:pre', todos los caracteres de espacio son afectados por el word-spacing.

En este ejemplo, el espacio entre cada palabra en los elementos H1 es aumentado en '1em'.

```
h1 { word-spacing: 1em }
```

16.5 Capitalización: propiedad 'text-transform'

'text-transform'

<i>Valor:</i>	capitalize uppercase lowercase none <u>inherit</u>
<i>Inicial:</i>	none
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Según lo especificado

Esta propiedad controla el efecto de la capitalización (cambios entre mayúsculas y minúsculas) del texto de un elemento. Los valores tienen los siguientes significados:

capitalize

Pone el primer carácter de cada palabra en mayúscula; el resto de caracteres no son afectados.

uppercase

Pone todos los caracteres de cada palabra en mayúscula.

lowercase

Pone todos los caracteres de cada palabra en minúsculas.

none

Ningún efecto de capitalización.

La transformación real en cada caso depende del lenguaje escrito. Vea RFC 3066 ([\[RFC3066 \]](#)) para las formas de encontrar el lenguaje de un elemento.

Las [aplicaciones del usuario con conformidad](#) pueden considerar el valor de `'text-transform'` como 'none' para los scripts escritos para los cuales no hay transformación.

En este ejemplo, todo el texto en un elemento H1 se transforma en mayúscula.

```
h1 { text-transform: uppercase }
```

16.6 Espacios en blanco: la propiedad 'white-space'

'white-space'

<i>Valor:</i>	normal pre nowrap pre-wrap pre-line <u>inherit</u>
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u>
<i>Valor computado:</i>	Según el especificado

Esta propiedad declara como son tratados los [espacios en blanco](#) dentro del elemento. Los valores tienen los siguientes significados:

normal

Este valor le indica a las aplicaciones del usuario cerrar las secuencias de espacios en blanco, y saltos de líneas como sea necesario para llenar las cajas de línea.

pre

Este valor impide a las aplicaciones del usuario cerrar las secuencias de espacios en blanco. Las líneas son cortadas solamente en las nuevas líneas del código fuente, al aparecer "\A" en el contenido generado.

nowrap

Este valor cierra los espacios en blanco como en 'normal', pero suprime los saltos de línea dentro del texto.

pre-wrap

Este valor impide a las aplicaciones del usuario cerrar las secuencias de espacios en blanco. Las líneas son cortadas en las nuevas líneas en el código fuente, o al aparecer "\A" en el contenido generado, y como sea necesario para llenar las cajas de línea.

pre-line

Este valor impide a las aplicaciones del usuario cerrar las secuencias de espacios en blanco. Las líneas son cortadas en las nuevas líneas en el código fuente, o al aparecer "\A" en el código generado, y como sea necesario para llenar las cajas de línea.

El siguiente ejemplo muestra que comportamiento de los espacios en blanco se espera para los elementos PRE y P, el atributo "nowrap" en HTML, y en el contenido generado.

```
pre      { white-space: pre }
p        { white-space: normal }
td[nowrap] { white-space: nowrap }
:before, :after { white-space: pre-line }
```

Además, el efecto de un elemento PRE de HTML con el atributo *non-standard* "wrap" es mostrado por el siguiente ejemplo:

```
pre[wrap] { white-space: pre-wrap }
```

16.6.1 El modelo de procesamiento de 'white-space'

Cualquier texto que directamente está contenido dentro de un bloque (no dentro de uno en línea) debería ser tratado como un elemento en línea anónimo.

Para cada elemento en línea (incluyendo en línea anónimos), se realizan los pasos siguientes, ignorando el procesamiento bidi de caracteres como si no estuvieran:

1. Cada carácter de espacio en blanco que no es un avance de línea que rodea un carácter de avance de línea (anterior y posterior) es eliminado si 'white-space' se fija como 'normal', 'no-wrap', o 'pre-line'.
2. Si 'white-space' es fijado a 'pre' o 'pre-wrap', cualquier secuencia de espacios (U+0020) no rota por un límite elemento es tratada como una secuencia de espacios sin saltos de línea (non-breaking). Sin embargo, existe una oportunidad de salto de línea al final de la secuencia.
3. Si 'white-space' es fijado a 'normal' o 'nowrap', los caracteres de avance de líneas son transformados para el procesado propuesto en uno de los siguientes caracteres: un carácter de espacio, un cero con carácter en blanco (U+200B), o sin carácter (es decir, no se procesa), de acuerdo a los algoritmos especificados de la AU-basados en el contenido del script.
4. Si 'white-space' es fijado a 'normal', 'nowrap', o 'pre-line',
 1. Todas las tabulaciones (U+0009) son convertidas a espacios (U+0020)
 2. cualquier espacio (U+0020) siguiente a otro espacio (U+0020) — hasta un espacio anterior en la línea, si ese espacio también tiene 'white-space' como 'normal', 'nowrap' o 'pre-line' — es eliminado.

Entonces, el bloque entero es procesado. Las líneas son presentadas, tomando bidi el nuevo reordenamiento en consecuencia, y envolviéndolo según lo especificado por la propiedad 'white-space'.

Cada línea es presentada,

1. Si un espacio (U+0020) al principio de la línea tiene 'white-space' fijado a 'normal', 'nowrap', o 'pre-line', es eliminado.

2. Todas las tabulaciones (U+0009) son procesadas como un movimiento horizontal que alinea el principio del límite del siguiente glyph con la siguiente parada de tabulación. Las paradas de tabulación ocurren en los puntos que son múltiplos de 8 espacios (una tabulación equivale a 8 espacios en blanco) (U+0020) procesados en la fuente del bloque desde comienzo del límite del contenido del bloque.
3. Si un espacio (U+0020) al final de una línea tiene 'white-space' fijado a 'normal', 'nowrap', o 'pre-line', es también eliminado.

16.6.2 Ejemplo de bidireccionalidad con espacios en blanco cerrados

Considerando el fragmento de margen siguiente, y prestando especial atención a los espacios (con fondos variado y bordes para énfasis e identificación):

```
<ltr>A <rtl>B </rtl>C </ltr>
```

...Donde el elemento `<ltr>` representa una fijación de izquierda a derecha y el elemento `<rtl>` representa una fijación de derecha a izquierda, y asumiendo que la propiedad 'white-space' es fijada como 'normal', el mencionado modelo de procesamiento resultaría como sigue:

- El espacio antes de B () se cerraría con el espacio después de A ().
- El espacio antes de C () se cerraría con el espacio antes de B ().

Esto dejaría dos espacios, uno después de the A en el nivel de fijación de izquierda a derecha, y uno después de B en el nivel de fijación de derecha a izquierda. Esto es entonces procesado de acuerdo al algoritmo bidireccional del Unicode, con el resultado final:

A BC

Observe que hay dos espacios entre A y B, y no entre B y C. Esto es evitado por utilizar el direccionamiento natural de caracteres en lugar del nivel de fijación explícito.

17 Tablas

Contenido

- [17.1 Introducción a las tablas](#)
- [17.2 El modelo de tabla de CSS](#)
 - [17.2.1 Objetos anónimos de la tabla](#)
- [17.3 Columnas](#)
- [17.4 Tablas en el modelo de formato visual](#)
 - [17.4.1 Posición y alineación del encabezado](#)
- [17.5 Composición visual del contenido de la tabla](#)
 - [17.5.1 Capas y transparencia en la tabla](#)
 - [17.5.2 Algoritmos para el ancho de la tabla: la propiedad 'table-layout'](#)
 - [Composición fija de la tabla](#)
 - [Composición automática de la tabla](#)
 - [17.5.3 Algoritmos para la altura de la tabla](#)
 - [17.5.4 Alineación horizontal en una columna](#)
 - [17.5.5 Efectos dinámicos en filas y columnas](#)
- [17.6 Bordos](#)
 - [17.6.1 El modelo de bordes separado](#)
 - [Bordes y fondos alrededor de las celdas vacías: la propiedad 'empty-cells'](#)
 - [17.6.2 El modelo de bordes cerrados](#)
 - [Resolución de conflictos de bordes](#)
 - [17.6.3 Estilos de bordes](#)

17.1 Introducción a las tablas

Una tabla puede ser utilizada para representar relaciones de tabulaciones entre datos. Los autores especifican estas relaciones en el [lenguaje del documento](#) y pueden especificar su *presentación* usando CSS 2.1.

En un medio visual, las tablas de CSS también se pueden utilizar para alcanzar composiciones específicas. En este caso, los autores no deben utilizar elementos relacionados a tablas en el lenguaje del documento, sino que deben aplicar el CSS a los elementos estructurales relevantes para alcanzar la composición deseada.

Los autores pueden especificar el formato visual de una tabla como rejilla rectangular de celdas. Las filas y las columnas de celdas se pueden organizar en grupos de filas y en grupos de columnas. Las filas, las columnas, los grupos de filas, los grupos de columnas, y las celdas pueden tener bordes dibujados alrededor de ellos (hay dos modelos de bordes en CSS 2.1). Los autores pueden alinear datos vertical u horizontalmente dentro de una celda y alinear datos en todas las celdas de una fila o de una columna.

Aquí hay una simple tabla descrita en HTML 4.0 de tres filas y tres columnas:

```
<TABLE>
<CAPTION>This is a simple 3x3 table</CAPTION>
<TR id="fila1">
  <TH>Header 1      <TD>Cell 1      <TD>Cell 2
```

```

<TR id="fila2">
  <TH>Header 2      <TD>Cell 3      <TD>Cell 4
<TR id="fila3">
  <TH>Header 3      <TD>Cell 5      <TD>Cell 6
</TABLE>

```

Este código crea una tabla (el elemento TABLE), tres filas (los elementos TR), tres celdas de título (los elementos TH), y seis celdas de datos (los elementos TD). Observe que las tres columnas de este ejemplo están especificadas implícitamente: hay tantas columnas en la tabla como se requiere para las celdas de los títulos y los datos.

La regla siguiente de CSS centra el texto horizontalmente en las celdas de título y presenta el texto en las celdas de título con un peso de fuente negrita.

```
th { text-align: center; font-weight: bold }
```

Las reglas siguientes alinean el texto de las celdas de título en su línea de base y centran verticalmente el texto en cada celda de datos:

```
th { vertical-align: baseline }
td { vertical-align: middle }
```

Las reglas siguientes especifican que la fila superior será rodeada por un borde azul sólido de 3px y cada una de las otras filas serán rodeadas por un borde negro sólido de 1px:

```
table { border-collapse: collapse }
tr#fila1 { border-top: 3px solid blue }
tr#fila2 { border-top: 1px solid black }
tr#row3 { border-top: 1px solid black }
```

Observe, sin embargo, que los bordes alrededor de las filas se superponen cuando las filas se juntan. ¿De qué color (negro o azul) y grosor (1px o 3px) será el borde entre fila1 y fila2? Discutiremos esto en la sección sobre [resolución de los conflictos de borde](#).

La regla siguiente pone el encabezado de la tabla encima de la tabla:

```
caption { caption-side: top }
```

Los ejemplos anteriores muestran como CSS trabaja con los elementos HTML 4.0; en HTML 4.0, la semántica de los distintos elementos de la tabla (TABLE, CAPTION, THEAD, TBODY, TFOOT, COL, COLGROUP, TH, y TD) está bien definidos. En otros lenguajes de documento (tales como aplicaciones XML), pueden no existir elementos de tabla predefinidos. Por lo tanto, CSS 2.1 permite que los autores "asignen" a los elementos del lenguaje del documento, elementos de la tabla a través de la propiedad '[display](#)'. Por ejemplo, la siguiente regla hace que el elemento FOO actúe como un elemento TABLE de HTML y el elemento BAR actúe como un elemento CAPTION:

```
FOO { display : table }
BAR { display : table-caption }
```


Discutiremos los diversos elementos de la tabla en la sección siguiente. En esta especificación, el termino elemento *de la tabla* se refiere a cualquier elemento implicado en la creación de una tabla. Un elemento de tabla "interno" es uno que produce una fila, un grupo de la filas, una columna, un grupo de la columnas, o una celda.

17.2 El modelo de tabla de CSS

El modelo de tabla de CSS se basa en el modelo de tabla de HTML 4.0, en la cual la estructura de una tabla es paralelo la disposición visual de la tabla. En este modelo, una tabla consiste en un encabezado opcional y cualquier número de filas de celdas. Se dice que el modelo de la tabla es "principalmente de filas" porque los autores especifican filas, no columnas, explícitamente en el lenguaje del documento. Las columnas son sacadas una vez las filas son especificadas – la primera explícitamente en la lengua de documento. Se derivan se han especificado las columnas una vez que todas las filas -- la primera celda de cada fila pertenece a la primera columna, la segunda a la segunda columna, etc.). Las filas y las columnas se pueden agrupar estructuralmente y esta agrupación se reflejado en la presentación (ej., puede dibujarse un borde alrededor de un grupo de filas).

Así, el modelo de tabla consiste en tablas, encabezados, filas, grupos de filas, columnas, grupos de la columnas, y celdas.

El modelo de CSS no requiere que el [lenguaje del documento](#) incluya los elementos que corresponden a cada uno de estos componentes. Para los lenguajes de documento (tales como aplicaciones XML) que no tienen elementos de tabla predefinidos, los autores deben asignar elementos del lenguaje de documento a los elementos de tabla; esto se hace con la propiedad `'display'`. Los valores siguientes para `'display'` asignan reglas de formato de la tabla a un elemento arbitrario:

table (En HTML: TABLE)

Especifica que un elemento define una tabla [a nivel de bloque](#): es un bloque rectangular que participa de un [contexto de formato de bloque](#).

Inline-table (En HTML: TABLE)

Especifica que un elemento define una tabla [a nivel de línea](#): es un bloque rectangular que participa de un [contexto de formato en línea](#).

table-row (En HTML: Tr)

Especifica que un elemento es una fila de celdas.

table-row-group (En HTML: TBODY)

Especifica que un elemento agrupa una o más filas.

table-header-group (En HTML: THEAD)

Como 'table-row-group', pero para el formato visual, el grupo de filas es siempre mostrado antes que el resto de filas y de grupos de filas y después de cualquier encabezado superior. Las aplicaciones del usuario que imprimen pueden repetir filas de encabezado en cada página ocupada por una tabla. El uso de múltiples elementos con `'display: table-header-group'` es indefinido.

table-footer-group (En HTML: TFOOT)

Como 'table-row-group', pero para el formato visual, el grupo de filas es siempre mostrado después de el resto de filas y de grupos de filas y antes de cualquier encabezado inferior. Las aplicaciones de usuarios que imprimen pueden repetir filas de pie en cada pagina ocupada por una tabla. El uso de múltiples elementos con `'display: table-footer-group'` es indefinido.

table-column (En HTML: COL)

Especifica que un elemento describe una columna de celdas.

table-column-group (En HTML: COLGROUP)

Especifica que un elemento agrupa una o más columnas.

table-cell (En HTML: TD, TH)

Especifica que un elemento representa una celda de la tabla.

table-caption (En HTML: CAPTION)

Especifica un encabezado para la tabla. El uso de múltiples elementos con 'display: caption' es indefinido, los autores no deben poner mas de un elemento con 'display:caption' dentro de un elemento de table o inline-table.

Los elementos 'display' 'table-column' o 'table-column-group' no son procesados (exactamente como si tuvieran 'display:none'), pero son útiles, porque pueden tener atributos que induzcan cierto estilo para las columnas que representan.

La [hoja de estilo predeterminada para HTML 4.0](#) en el apéndice, ilustra el uso de estos valores para HTML 4.0:

```
table      { display: table }
tr        { display: table-row }
thead     { display: table-header-group }
tbody     { display: table-row-group }
tfoot     { display: table-footer-group }
col       { display: table-column }
colgroup  { display: table-column-group }
td, th    { display: table-cell }
caption   { display: table-caption }
```

Las aplicaciones del usuario pueden ignorar estos valores para la propiedad 'display' de los elementos de tabla de HTML, puesto que las tablas de HTML se pueden procesar usando otros algoritmos previstos para la representación compatible inversa. Sin embargo, esto no significa que se desaliente el uso de 'display: table' en otros, elementos no tablas en HTML.

17.2.1 Objetos anónimos de la tabla

Los lenguajes de documento con excepción del HTML pueden no contener todos los elementos del modelo de tabla de CSS 2.1. En estos casos, los elementos que "faltan" se deben asumir en orden para que el modelo de tabla funcione. Cualquier elemento de tabla generará automáticamente los objetos de tabla anónimos necesarios alrededor de sí mismo, consistiendo por lo menos en tres objetos jerarquizados que corresponden a un elemento 'table'/'inline-table', a un elemento 'table-row', y a un elemento 'table-cell'. Los elementos que falta generan objetos anónimos (ej., cajas anónimas en la composición visual de la tabla) según las reglas siguientes:

1. Si el padre P de un elemento 'table-cell' T no es un 'table-row', un objeto correspondiente a 'table-row' será generado entre P y T. Este objeto abarcará todos los hermanos 'table-cell' consecutivos (en la estructura del documento) de T.
2. Si el padre P de un elemento 'table-row' T no es un elemento 'table', 'inline-table', 'table-header-group', 'table-footer-group' o 'table-row-group', un objeto correspondiente al elemento 'table' será generado entre P y T. Este objeto abarcará todos los hermanos consecutivos (en la estructura del documento) de T que requieran un padre 'table': 'table-row', 'table-row-group', 'table-header-group', 'table-footer-group', 'table-column', 'table-

- column-group', y 'table-caption'. T y los hermanos de T pueden ser también objetos 'table-row' anónimos generados por la regla 1.
3. Si el padre P de un elemento 'table-column' T no es un elemento 'table', 'inline-table', o 'table-column-group', un objeto correspondiente al elemento 'table' será generado entre P y T. Este objeto contendrá todos los hermanos consecutivos (en la estructura del documento) de T que requieran un padre 'table': 'table-row', 'table-row-group', 'table-header-group', 'table-footer-group', 'table-column', 'table-column-group', y 'table-caption', incluyendo algún objeto 'table-row' anónimo generado por la regla 1.
 4. Si el padre P de un elemento 'table-row-group' (o 'table-header-group', 'table-footer-group', o 'table-column-group' o 'table-caption') T no es un 'table' o 'inline-table', un objeto correspondiente al elemento 'table' será generado entre P y T. Este objeto contendrá todos los hermanos consecutivos (en la estructura del documento) de T que requieran un padre 'table': 'table-row', 'table-row-group', 'table-header-group', 'table-footer-group', 'table-column', 'table-column-group', y 'table-caption', incluyendo cualquier objeto 'table-row' anónimo generado por la regla 1.
 5. Si un hijo T de un elemento 'table' (o 'inline-table') P no es un elemento 'table-row-group', 'table-header-group', 'table-footer-group', 'table-caption', 'table-column', 'table-column-group' o 'table-row', un objeto correspondiente a un elemento 'table-row' será generado entre P y T. Este objeto contendrá todos los hermanos consecutivos de T que no sean 'table-row-group', 'table-header-group', 'table-footer-group', 'table-caption', 'table-column', 'table-column-group' o 'table-row' elements.
 6. Si un hijo T de un elemento 'table-row-group' (o 'table-header-group' o 'table-footer-group') P no es un elemento 'table-row', un objeto que corresponde a un elemento 'table-row' será generado entre P y T. Este objeto abarcará a todos los hermanos de T que no son elementos 'table-row'.
 7. Si un hijo T de un elemento 'table-row' no es un elemento 'table-cell', un objeto correspondiente a un elemento 'table-cell' será generado entre P y T. Estos objetos abarcaran a todos los hermanos consecutivos de T que no sean elementos 'table-cell'.

En este ejemplo XML, un elemento 'table' se asume para contener al elemento HBOX:

```
<HBOX>
  <VBOX>George</VBOX>
  <VBOX>4287</VBOX>
  <VBOX>1998</VBOX>
</HBOX>
```

Porque la hoja de estilo asociada es:

```
HBOX { display: table-row }
VBOX { display: table-cell }
```

En este ejemplo, tres elementos 'table-cell' son asumidos para contener el texto en cada FILA. Observe que el texto es además encapsulado en cajas en línea anónimas, como se explica en el modelo de formato visual:

```
<STACK>
  <ROW>Esta es la fila <D>superior</D>. </ROW>
  <ROW>Esta es la fila <D>del medio</D>. </ROW>
  <ROW>Esta es la fila <D>inferior</D>. </ROW>
</STACK>
```

La hoja de estilo es:

```
STACK { display: inline-table }
ROW   { display: table-row }
D     { display: inline; font-weight: bolder }
```

17.3 Columnas

Las celdas de la tabla pueden pertenecer a dos contextos: filas y columnas. Sin embargo, en código fuente del documento las celdas son descendentes de las filas, nunca de las columnas. No obstante, algunos aspectos de las celdas pueden ser influenciados poniendo propiedades a las columnas.

Las propiedades siguientes se aplican a los elementos columna y grupo de columnas:

'border'

Las distintas propiedades de borde se aplican a las columnas solo si 'border-collapse' es definido como 'collapse' sobre el elemento tabla. En ese caso, los bordes definidos para las columnas y grupos de columnas son introducidos al [algoritmo de resolución de conflictos](#) que elige el estilo del borde para cada lado del borde de la celda.

'background'

Las propiedades background determinan el fondo para las celdas en la columna, pero solo si tanto la celda y la fila tiene fondos transparentes. Ver "[Capas y transparencia de la tabla](#)"

'width'

La propiedad 'width' determina el ancho mínimo de la columna.

'visibility'

Si el 'visibility' de una columna es puesto como 'collapse', ninguna de las celdas en la columna son procesadas, y las celdas contenidas dentro de otras columnas son recortadas. Además, el ancho de la tabla es disminuido por el ancho de la columna que se hubiera quitado. Ver "[efectos dinámicos](#)" abajo. Otros valores para 'visibility' no tienen efecto.

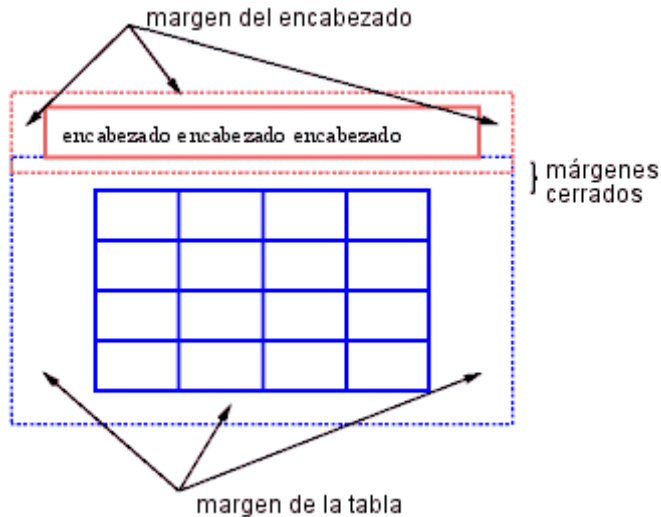
Aquí hay algunos ejemplos de reglas de estilo que fijan propiedades de columnas. Las primeras dos reglas juntas implementan el atributo "rules" de HTML 4.0 con un valor de "cols". La tercera regla hace que la columna "totales" sea azul, las dos reglas finales muestran como hacer una columna de tamaño fijo, usando el [algoritmo de estructura fija](#).

```
col    { border-style: none solid }
table  { border-style: hidden }
col.totales { background: blue }
table  { table-layout: fixed }
col.totales { width: 5em }
```

17.4 Las tablas en el modelo de formato visual

En los términos del [modelo de formato visual](#), una tabla debe funcionar como un elemento [a nivel de bloque](#) o [a nivel de línea](#). Las tablas tienen contenido, relleno, bordes y márgenes.

En ambos casos, los elementos de la tabla generan una caja [anónima](#) que contiene a la propia caja de la tabla y a la caja del encabezado (si está presente). Las cajas de tabla y de encabezados retienen sus propias áreas de contenido, relleno, márgenes y bordes, y las dimensiones de la caja rectangular anónima son las mínimas requeridas para contener a ambas. Los márgenes verticales se cierran donde la caja de la tabla y la caja del encabezado se tocan. Cualquier reposicionamiento de la tabla debe mover la caja anónima entera, no solo la caja de la tabla, de modo que el encabezado siga a la tabla.



[D]

Diagrama de una tabla con un encabezado encima de ella; el margen inferior del encabezado se cierra con el margen superior de la tabla.

17.4.1 Posición y alineación del encabezado

'caption-side'

Valor: top | bottom | inherit
Inicial: top
Se aplica a: Elementos 'table-caption'
Se hereda: Si
Porcentajes: N/A
Medio: visual
Valor computado: Según el especificado

Esta propiedad especifica la posición de la caja del encabezado con respecto a la caja de la tabla. Los valores tiene los siguientes significados:

top

Posiciona la caja del encabezado encima de la caja de la tabla.

bottom

Posiciona la caja del encabezado abajo de la caja de la tabla.

Los encabezados encima o debajo del elemento 'table' son estructurados mucho más como si fueran un elemento de bloque antes y después de la tabla, excepto que (1) hereden las propiedades heredables de la tabla, y (2) no se consideran como una caja de bloque para el propósito de ningún elemento 'run-in' que pueda preceder a la tabla.

Un encabezado que está arriba o abajo de una caja de tabla también se comporta como una caja de bloque para el cálculo del ancho y el alto; el ancho y alto son calculados con respecto al bloque de contención de la caja de la tabla.

Para alinear el contenido del encabezado horizontalmente dentro de la caja del encabezado, se utiliza la propiedad 'text-align'.

En este ejemplo, la propiedad 'caption-side' coloca los encabezados debajo de las tablas. El encabezado será tan ancho como el padre de la tabla, y el texto del encabezado estará justificado a la izquierda.

```
caption { caption-side: bottom;
          width: auto;
          text-align: left }
```

17.5 Composición visual del contenido de las tablas

Los elementos internos de la tabla generan cajas rectangulares con contenido y bordes. Las celdas tienen relleno también. Los elementos internos de la tabla no tienen márgenes.

La composición visual de estas cajas está gobernada por una rejilla rectangular, irregular de filas y columnas. Cada caja ocupa un número entero de celdas de la rejilla, determinado de acuerdo a las siguientes reglas. Estas reglas no se aplican a HTML 4 o versiones anteriores de HTML; HTML impone sus propias limitaciones al alcance de las filas y columnas.

1. Cada caja de fila ocupa una fila de celdas de la rejilla. Juntas, las cajas de filas llenan la tabla de arriba hacia abajo en el orden en que aparecen en el documento fuente (es decir, la tabla ocupa exactamente tantas filas de la rejilla como elementos fila existen).
2. Un grupo de filas ocupa las mismas celdas de la rejilla como filas contiene.
3. Una caja de columna ocupa una o más columnas de la rejilla de celdas. Las cajas de columna son colocadas una al lado de la otra en el orden en que aparecen. La primera caja de columna puede estar a la izquierda o a la derecha, dependiendo del valor de la propiedad 'direction' de la tabla.
4. Una caja de un grupo de columnas ocupa las mismas celdas de la rejilla como columnas contiene.
5. Las celdas pueden contener varias filas o columnas. (Aunque CSS2.1 no define como se determina el número de filas y columnas abarcadas, una aplicación del usuario puede tener un conocimiento especial acerca del documento fuente, una versión futura de CSS puede proporcionar una manera de expresar este conocimiento en la sintaxis de CSS) Cada celda es, en consecuencia, una caja rectangular, o una o más celdas de la rejilla de ancho y de alto. La fila superior de este rectángulo es la fila especificada por el padre de la celda. El rectángulo debe estar tan a la izquierda como sea posible, pero no puede superponerse con ninguna otra caja de celda, y debe estar a la derecha de todas las celdas de la misma fila que aparecen antes en el documento fuente. (Esta restricción se

mantiene si la propiedad 'direction' de la tabla es 'ltr'; si 'direction' es 'rtl', intercambie "left" and "right" en la frase anterior)

- Una caja de celda no puede extenderse mas allá de la ultima caja de fila de una tabla o grupo de filas (row-group); las ampliaciones del usuario deben acortarla hasta que se ajuste.

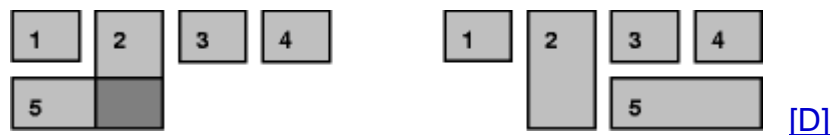
Nota. Las celdas de la tabla pueden ser posicionadas, pero no es recomendable: el posicionamiento absoluto y fijo (fixed), como también el flotante, eliminan una caja del flujo, afectando al tamaño de la tabla.

Aquí hay dos ejemplos. El perímetro supone que aparece en un documento HTML, el segundo en un documento XHTML:

```
<TABLE>
<TR><TD>1 <TD rowspan="2">2 <TD>3 <TD>4
<TR><TD colspan="2">5
</TABLE>
```

```
<table>
<tr><td>1 </td><td rowspan="2">2 </td><td>3 </td><td>4 </td></tr>
<tr><td colspan="2">5 </td></tr>
</table>
```

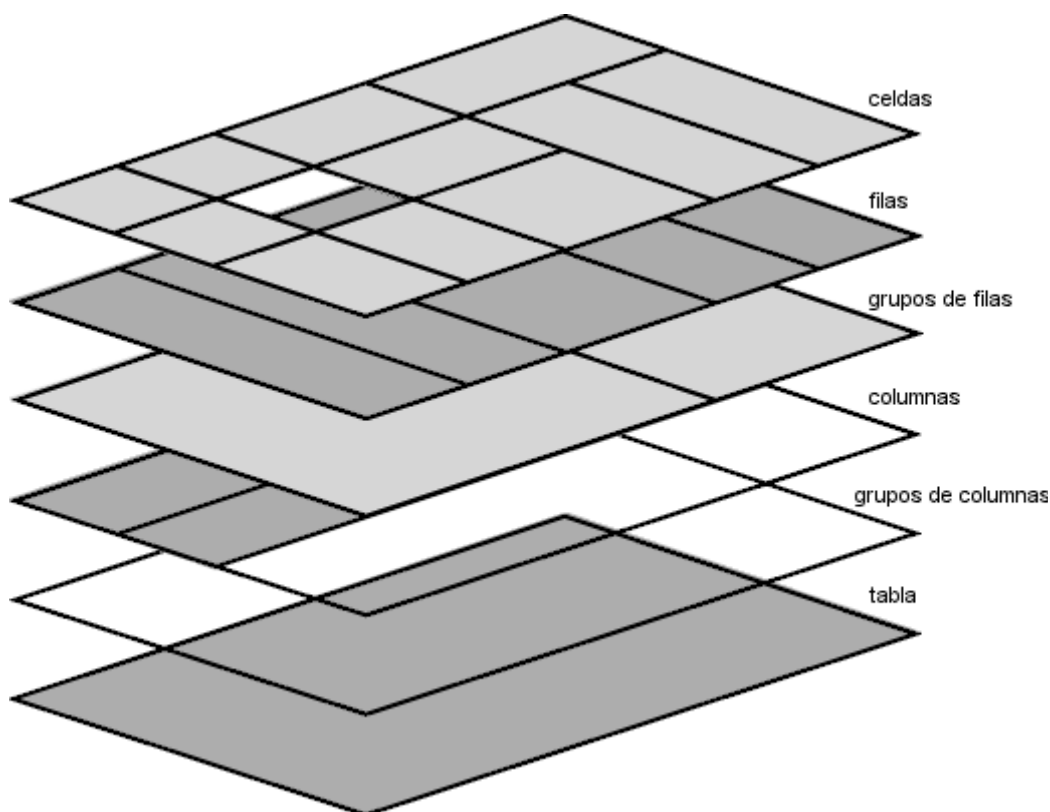
La segunda tabla es estructurada como en la figura de la derecha. Sin embargo, el procesamiento de las tablas en HTML está explícitamente indefinido por HTML, y CSS no trata de definirlo. Las aplicaciones del usuario son libres de procesarlas, por ejemplo como en la figura de la izquierda.



A la izquierda, un posible procesamiento de una tabla errónea en HTML 4.0; a la derecha, el único formato posible de una tabla similar a XHTML.

17.5.1 Capas y transparencia en la tabla

Con el propósito de encontrar el fondo de cada celda de la tabla, los distintos elementos de la tabla pueden considerarse como si estuvieran en seis capas superpuestas. El fondo aplicado sobre un elemento en una de estas capas solo será visible si las capas encima de ella tienen un fondo transparente.



[D]

Esquema de las capas de la tabla.

1. La capa inferior es un simple plano, representando la caja de la tabla en si. Como todas las cajas, puede ser transparente.
2. La capa siguiente contiene los grupos de columnas. Cada grupo de columna se extiende desde la parte superior de las celdas en la fila superior hasta la inferior de las celdas en la fila inferior y desde el borde izquierdo de la columna mas a la izquierda hasta el borde derecho de la columna mas a la derecha. El fondo se extiende hasta cubrir todo el área de todas las celdas que provienen del grupo de columnas, pero esta extensión no afecta al posicionado de la imagen de fondo.
3. Sobre los grupos de columna están las áreas representando las cajas de las columnas. Cada columna es tan alta como el grupo de columnas y tan amplia como una celda normal (single-column-spanning) en la columna. El fondo se extiende hasta cubrir todo el área de todas las celdas que provienen de la columna, incluso si su contenido sobresale de la columna, pero esta extensión no afecta al posicionamiento de la imagen de fondo.
4. La siguiente es la capa que contiene los grupos de filas. Cada grupo de filas se extiende desde la esquina superior izquierda de su celda mas arriba en la primera columna hasta el borde inferior derecho de su celda más abajo en la ultima columna.
5. La penúltima capa contiene las filas. Cada fila es tan amplia como el grupo de filas y tan alta como una celda normal (single-row-spanning) cell en el flujo. Como con las columnas, el fondo se extiende hasta cubrir todo el área de todas las celdas que son originadas en la fila, incluso si su contenido sobresale de la fila, pero esta extensión no afecta al posicionamiento de la imagen de fondo.
6. La capa superior contiene las propias celdas. Como muestra el dibujo, aunque todas las filas contienen el mismo numero de celdas, puede que no todas las celdas tengan especificado un contenido. Si el valor de su propiedad 'empty-cells' es 'hide' estas celdas

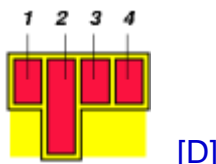
"vacías" son transparentes dejando ver el fondo de la cell, la fila, el grupo de filas, la columna and grupo de columnas, mostrando el fondo de la tabla.

Los límites de las filas, columnas, grupos de filas and grupos de columnas en el [modelo de bordes cerrados](#) coincide con las hipotéticas líneas de la rejilla en la que los bordes de las celdas son centradas. (y así, en este modelo, las filas juntas cubren exactamente la tabla, no dejando ningún hueco, lo mismo con las columnas) en el [modelo de bordes separados](#), el límite coincide con el [límite del borde](#) de las celdas. (y así, en este modelo puede haber huecos entre las filas, las columnas, los grupos de filas o los grupos de columnas, correspondiendo a la propiedad '[border-spacing](#)')

En el siguiente ejemplo, la primera fila contiene cuatro celdas, pero la segunda fila no contiene celdas, y así el fondo de la tabla se distingue a través de esta, excepto donde una celda de la primera fila alcanza hasta esa fila. El siguiente código HTML y reglas de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Table example</TITLE>
    <STYLE type="text/css">
      TABLE { background: #fff0; border-collapse: collapse;
        empty-cells: hide }
      TD { background: red; border: double black }
    </STYLE>
  </HEAD>
  <BODY>
    <TABLE>
      <TR>
        <TD> 1
        <TD rowspan="2"> 2
        <TD> 3
        <TD> 4
      </TR>
      <TR><TD></TD></TR>
    </TABLE>
  </BODY>
</HTML>
```

Podría tener el siguiente formato:



El diagrama muestra una tabla con dos filas y cuatro columnas. Las columnas están etiquetadas con los números 1, 2, 3 y 4. La primera fila contiene cuatro celdas: la primera es roja y contiene el número '1', la segunda es roja y contiene el número '2', la tercera es roja y contiene el número '3', y la cuarta es roja y contiene el número '4'. La segunda fila contiene una sola celda roja que abarca las columnas 1 y 2, y contiene el número '2'. Las celdas en las columnas 3 y 4 de la segunda fila están vacías, mostrando el fondo blanco de la tabla. El fondo de la tabla es amarillo.

Tabla con tres celdas vacías en la fila inferior.

Observe que si la tabla tiene '[border-collapse: separate](#)', el fondo del área dada por la propiedad '[border-spacing](#)' es siempre el fondo del elemento table. Ver el [modelo de bordes separados](#).

17.5.2 Algoritmos para el ancho de la tabla: la propiedad **'table-layout'**

CSS no define una composición "óptima" para las tablas porque, en muchos casos, lo óptimo es una cuestión de gustos. CSS no define restricciones que las aplicaciones del usuario deban respetar, y son libre de preferir un procesamiento rápido sobre uno preciso, excepto cuando se selecciona el "algoritmo de composición fija".

Observe que esta sección anula las reglas que se aplican para calcular los anchos como se describe en la [sección 10.3](#). En particular, si los márgenes de la tabla son fijados a '0' y el ancho a 'auto', la tabla no pondrá el tamaño automáticamente para llenar su bloque de contención. Sin embargo, una vez el valor computado de 'width' para la tabla es encontrado (usando los algoritmos dados abajo o, cuando sea necesario, algún otro algoritmo dependiente de la AU) entonces otras partes de la sección 10.3 se aplican. Por otro lado la tabla puede ser centrada usando los márgenes izquierda y derecha como 'auto', por ejemplo.

Futuras versiones de CSS pueden introducir los modos de hacer tablas automáticamente encajando sus bloques de contención.

'table-layout'

Valor: auto | fixed | inherit
Inicial: auto
Se aplica a: Elementos 'table' e 'inline-table'
Se hereda: no
Porcentajes: N/A
Medio: visual
Valor computado: Según el especificado

La propiedad **'table-layout'** controla el algoritmo utilizado para componer las celdas, filas y columnas de la tabla. Los valores tienen los significados siguientes:

fixed

Se utiliza el algoritmos de composición fija de tabla.

auto

Se utiliza cualquier algoritmo de composición de tablas

Los dos algoritmos son descritos abajo.

Composición fija de la tabla

Con este (rápido) algoritmo, la composición horizontal de la tabla no depende del contenido de las celdas, solo depende del ancho de la tabla, el ancho de las columnas, y los bordes o espacio entre las celdas.

El ancho de la tabla puede ser especificado explícitamente con la propiedad **'width'**. Un valor de 'auto' (tanto para 'display: table' como para 'display: inline-table') significa que se utiliza el algoritmo de [composición automática de la tabla](#). Sin embargo, si la tabla es un bloque a nivel de tabla ('display: table') en el flujo normal, la AU puede (pero no tiene porque) utilizar el algoritmo de [10.3.3](#) para procesar el ancho y aplicarlo en la composición fija de la tabla incluso si el ancho es 'auto'.

Si la AU soporta composición fija de tabla cuando 'width' es 'auto', lo siguiente creará una tabla que es 4em mas estrecha que el bloque de contención:

```
table { table-layout: fixed;
        margin-left: 2em;
        margin-right: 2em }
```

En el algoritmo de composición fija de la tabla, el ancho de cada columna es determinado como sigue:

1. Un elemento columna con un valor distinto a 'auto' para la propiedad 'width' determina el ancho de esa columna.
2. Si no, una celda en de la primera fila con un valor distinto de 'auto' para la propiedad 'width' determina el ancho para esa columna. Si la celda abarca mas de una columna, el ancho se divide entre las columnas.
3. Las columnas restantes se dividen el espacio horizontal restante de la tabla (menos los borde o el espacio entre las celdas).

El ancho de la tabla es entonces el mayor de los valor de la propiedad 'width' para el elemento tabla y la suma de los anchos de las columnas (mas el espacio entre celdas o bordes). Si la tabla es mas amplia que las columnas, el espacio extra debe ser distribuido entre las columnas.

No se define que ocurre si una fila subsecuente tiene más columnas que la primera. Cuando se utiliza 'table-layout: fixed', los autores no deberán omitir las columnas desde la primera fila.

De esta manera, la aplicación del usuario puede comenzar a construir la tabla una vez que ha recibido la primera fila completa. Las celdas de las filas subsecuentes no afectan al ancho de la columna. Cualquier celda con un contenido que la desborda utiliza la propiedad 'overflow' para determinar si recorta el contenido desbordado.

Composición automática de la tabla

En este algoritmo (que generalmente no requiere más de dos pasos), el ancho de la tabla está dado por el ancho de sus columnas (y los [bordes](#) que intervienen). Este algoritmo refleja el comportamiento de varias aplicaciones del usuario de HTML populares al escribirse esta especificación. Las AUs no necesitan implementar este algoritmos para determinar la composición de la tabla en el cado de 'table-layout' como 'auto'; pueden utilizar cualquier otro algoritmo.

Este algoritmo puede resultar ineficiente dado que requiere a las ampliaciones del usuario tener acceso a todo el contenido de la tabla antes de determinar la composición final y puede demandar más de una pasada.

El ancho de las columnas se determina como sigue:

1. Se calcula el ancho mínimo del contenido (AMC) de cada celda: el contenido estructurado puede abarcar cualquier numero de líneas pero no puede desbordar la caja de la celda. Si el 'ancho' (A) especificado de la celda es mayor que AMC, A es el ancho mínimo de la celda. Un valor 'auto' significa que AMC es el ancho mínimo de la celda.

Además, se calcula el ancho "máximo" de celda de cada celda cell: se da formato al contenido sin saltos de línea salvo donde aparecen los saltos de línea explícitos.

2. Por cada columna, se determina un ancho máximo y mínimo de columna para las celdas que ocupan solo esa columna. El mínimo es el requerido por la celda con el mayor ancho mínimo de celda (o el 'ancho' de la columna, cualquiera que sea mayor). El máximo es el requerido por la celda con mayor ancho máximo de celda (o el 'ancho' de la columna, cualquiera que sea mayor).
3. Por cada celda que abarca más de una columna, se incrementan los anchos mínimos de las columnas que abarca de modo que juntas, estas al menos tan anchas como la celda. Se hace lo mismo para los anchos máximo. Si es posible, se ensancha las columnas abarcadas en aproximadamente el mismo valor.

Esto da un ancho máximo y mínimo para cada columna. Los anchos de las columnas influyen en el ancho final de la tabla como sigue:

1. Si la propiedad 'width' del elemento 'table' o 'inline-table' tiene un valor computado (A) distinto de 'auto', el valor de la propiedad utilizado para la composición es el mayor entre A y el ancho mínimo requerido por todas las columnas más el espacio entre las celdas o bordes (MIN). Si A es mayor que MIN, el ancho extra deberá ser distribuido entre las columnas.
2. Si el elemento 'table' o 'inline-table' tiene 'width: auto', el ancho de la tabla utilizado para la composición es el mayor entre el ancho del bloque de contención de las tablas y MIN. Sin embargo, si el ancho máximo requerido por las columnas más el espacio entre las celdas o bordes (MAX) es menor que el bloque de contención, se usa MAX.

Un valor de porcentaje para un ancho de columna es relativo al ancho de la tabla. Si la tabla tiene 'width: auto', un porcentaje representa una condición en el ancho de la columna, la que una AU deberá tratar de satisfacer. (Obviamente, esto no siempre es posible: si el ancho de la columna es '110%', la condición no puede satisfacerse.)

Nota. *En este algoritmo, tanto las filas (y grupos de filas) y las columnas (y grupos de columnas) restringen y son restringidas por las dimensiones de las celdas que contienen. Colocando el ancho a una columna se puede influir indirectamente en la altura de una fila, y viceversa.*

17.5.3 Algoritmo para la altura de la tabla

La altura de la tabla está dada por la propiedad 'height' del elemento 'table' o 'inline-table'. Un valor de 'auto' significa que la altura es la suma de las alturas de las filas más cualquier espacio entre celdas o bordes. Cualquier otro valor especifica la altura explícitamente; la tabla puede resultar de ese modo más larga o más corta que la altura de sus filas. CSS2.1 no especifica un procesamiento cuando la altura especificada de la tabla difiere de la altura del contenido, en particular si la altura del contenido debe sustituir la altura especificada; si no, como debe distribuirse el espacio extra entre las filas que sumadas resultan menores que la altura especificada para la tabla; o, si la altura del contenido excede la altura especificada de la tabla, si la AU debe proporcionar un mecanismo de desplazamiento. **Nota.** Futuras versiones de CSS pueden especificar esto mejor.

La altura de la caja del elemento 'table-row' se calcula una vez que la aplicación del usuario tiene disponible todas las celdas de la fila: es la máxima 'altura' especificada para la fila y la mínima altura (MIN) requerida por las celdas. Un valor de 'height' como 'auto' para un 'table-row' significa que la altura de la fila utilizada para la composición es MIN. MIN depende de la caja de la celda y de la alineación de la caja de la celda de (muy parecido al calculo de la altura de una [caja de línea](#)). CSS2.1 no define a que se refieren los valores de porcentajes en 'height' cuando se especifican para las filas o grupos de filas de la tabla.

En CSS2.1, la altura de una caja de celda es el máximo de la propiedad 'height' de las celdas de la tabla y la mínima altura requerida por el contenido (MIN). Un valor 'auto' para 'height' implica que el valor MIN será usado para la composición. CSS2.1 no especifica a que se refieren los valores de porcentaje en 'height' cuando se especifican para las celdas de la tabla.

CSS2.1 no especifica como las celdas que abarcan más de una fila afectan al calculo de la altura de la fila excepto que la suma de las alturas de las filas involucradas debe ser suficiente para contener a la celda que abarca las filas.

La propiedad 'vertical-align' de cada celda determina su alineación dentro de la fila. Cada contenido de cada celda tienen una línea base, una parte superior, una parte media y una parte inferior, como la propia fila. En el contexto de las tablas, los valores para 'vertical-align' tiene los siguientes significados:

baseline

La línea base de la celda se pone a la misma altura que la línea base de las primeras filas que abarca (ver abajo para la definición de líneas de base de celda y filas).

top

La parte superior de la caja es alineada con la parte superior de la primera fila que contiene.

bottom

La parte inferior de cada caja de la celda es alineada con la parte inferior de la ultima fila que contiene.

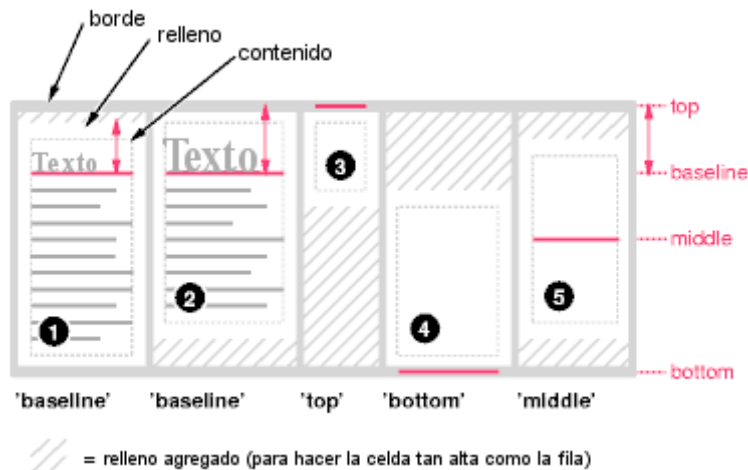
middle

El centro de la celda es alineado con el centro de las filas que contiene.

sub, super, text-top, text-bottom, <medida>, <porcentaje>

Estos valores no se aplican a las celdas, en cambio la celda se alinea por la línea de base.

La línea de base de una celda es la línea de base de la primera [caja de línea](#) en la celda. Si no hay texto, la línea de base es la línea de base de cualquier objeto mostrado en la celda, o, si no hay ninguno, la parte inferior de la caja de la celda. La distancia máxima entre la parte superior de la caja de la celda y la línea de base en todas las celdas que tengan 'vertical-align: baseline' es utilizada para fijar la línea base de la fila. Aquí hay un ejemplo:



[D]

Diagrama mostrando el efecto de los distintos valores de 'vertical-align' sobre las celdas de la tabla.

Las cajas de las celdas 1 y 2 están alineadas por sus líneas bases. La caja de la celda 2 tiene mayor altura por encima de la línea base, de modo que determina la línea base de la fila. Observe que si no hay caja de celda alineada a su línea base, la fila no tendrá (no necesitará) una línea de base.

Para evitar situaciones ambiguas, la alineación de las celdas sigue en el siguiente orden:

1. Primero las celdas que son alineadas sobre su línea base son posicionadas. Esto establecerá la línea base de la fila. después se posicionan las celdas con 'vertical-align: top'.
2. La fila ahora tiene una parte superior, posiblemente una línea de base, y una altura provisional, que es la distancia desde la parte superior hasta la parte más baja de las celdas posicionadas hasta el momento. (Ver las condiciones sobre el relleno de la celda más abajo.)
3. Si alguna de las celdas restantes, aquellas alineadas en la parte inferior o a la mitad, tiene una altura mayor que la actual altura de la fila, la altura de la fila será incrementada al máximo de aquellas celdas, bajando la parte inferior.
4. Finalmente se posicionan las celdas restantes.

Las cajas de celdas que son más pequeñas que la altura de la fila reciben relleno superior e inferior extra.

17.5.4 Alineación horizontal en una columna

La alineación horizontal del contenido de una celda dentro de una caja de celda se especifica con la propiedad 'text-align'.

17.5.5 Efectos dinámicos en filas y columnas

La propiedad 'visibility' toma el valor 'collapse' para filas, grupos de filas, columnas, y grupos de columna. Este valor provoca que la fila o columna entera se deje de mostrar, y el espacio normalmente ocupado por la fila o columna queda disponible para otro contenido. El contenido de las filas y columnas que abarca que se cruzan cerrando columnas y filas son acortadas. La supresión de la fila o columna, sin embargo, no afecta de ningún modo a la composición de la tabla. Esto permite efectos dinámicos al quitar filas de tablas o columnas sin forzar una recomposición de la tabla para calcular el cambio potencial en los requerimientos de la columna.

17.6 Bordos

Hay dos modelos distintos para definir los bordes de las celdas de una tabla en CSS. Uno es mas conveniente para los llamados bordes separados alrededor de las celdas individuales, el otro es conveniente para bordes continuos de una punta a otra de la tabla. Muchos estilos de bordes pueden conseguirse con cualquier modelo, de modo que es a menudo una cuestión de gustos cual de ellos se utiliza.

'border-collapse'

Valor: collapse | separate | inherit
Inicial: separate
Se aplica a: Elementos 'table' y 'inline-table'
Se hereda: Si
Porcentaje: N/A
Medio: visual
Valor computado: Según el especificado

Esta propiedad selecciona un modelo de borde de la tabla. El valor 'separate' selecciona el modelo de bordes separado. El valor 'collapse' selecciona el modelo de bordes cerrado. Los modelos son descritos abajo.

17.6.1 El modelo de bordes separado

'border-spacing'

Valor: <medida> <medida>? | inherit
Inicial: 0
Se aplica a: Elementos 'table' y 'inline-table' elements*
Se hereda: Si
Porcentajes: N/A
Medio: visual
Valor computado: Las dos medidas absolutas

*) Nota: Las aplicaciones de usuario pueden también aplicar la propiedad 'border-spacing' a elementos 'frameset'. Que elementos son elementos 'frameset' no se define en esta especificación y es definido por el lenguaje del documento. Por ejemplo, HTML4 define un elemento <FRAMESET>, y XHTML 1.0 define un elemento <frameset>. La propiedad 'border-spacing' de un elemento 'frameset' puede ser así utilizada como un valor sustituto para el atributo no estándar 'framespacing'.

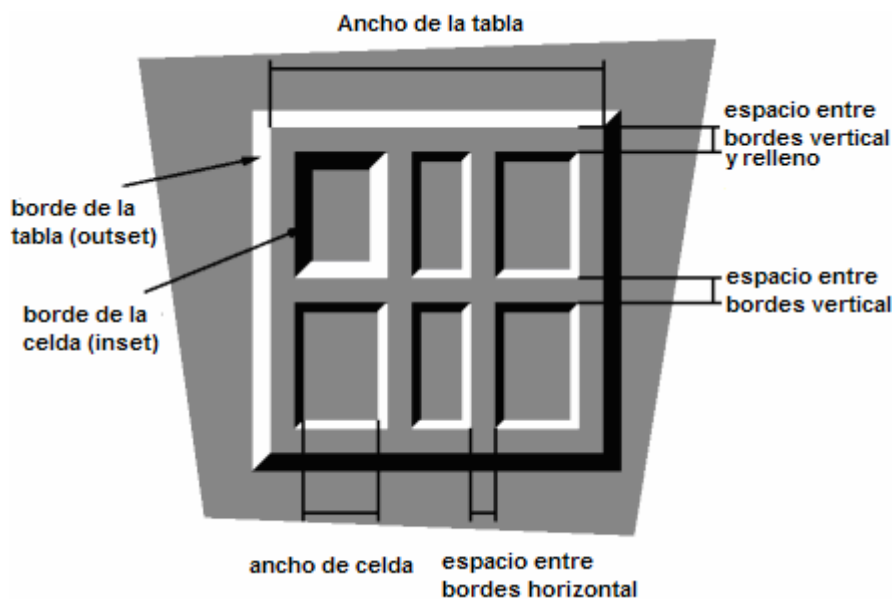
Las medidas especifican la distancia que separa los bordes de las celdas adyacentes. Si una medida es especificada, está da el espacio horizontal y vertical. Si dos medidas se especifican, la primera da el espacio horizontal y la segunda el espacio vertical. Las medidas no pueden ser negativas.

La distancia entre el borde de la tabla y los bordes de las celdas en el límite de la tabla es el relleno para ese lado, más la distancia de espacio del borde relevante. Por ejemplo, en el lado derecho, la distancia del relleno derecho + es espacio del borde horizontal.

En este modelo, cada celda tiene un borde individual. La propiedad 'border-spacing' especifica la distancia entre los bordes de celdas adyacentes. En este espacio, el fondo de la fila, la columna, el grupo de filas, y el grupo de columnas es visible, permitiendo mostrar a través del mismo el fondo de la tabla. Las filas, columnas, grupos de filas y grupos de columnas no pueden tener bordes (es decir, las aplicaciones del usuario deben ignorar las propiedades de bordes para estos elementos).

La tabla en la figura de abajo podría ser el resultado de una hoja de estilo como esta:

```
table      { border: outset 10pt;
             border-collapse: separate;
             border-spacing: 15pt }
td         { border: inset 5pt }
td.special { border: inset 10pt } /* The top-left cell */
```



[D]

Una tabla con 'border-spacing' fijada con un valor de medida. Observe que cada celda tiene su propio borde, y la tabla tiene también el borde separado.

Bordes y fondos alrededor de celdas vacías: la propiedad 'empty-cells'

'empty-cells'

Valor:	show hide <u>inherit</u>
inicial:	show
Se aplica a:	Elementos 'table-cell'
Se hereda:	Si
Porcentajes:	N/A
Medio:	<u>visual</u>
Valor computado:	Según el especificado

En el método de bordes separados, esta propiedad controla el procesamiento de bordes y fondos alrededor de las celdas que no tienen contenido visible. Las celdas vacías con la propiedad 'visibility' fijadas como 'hidden' se consideran que no tienen un contenido visible. El contenido visible incluye " " y otro espacio en blanco excepto ASCII CR ("\0D"), LF ("\0A"), tab ("\09"), y espacio ("\20").

Cuando esta propiedad tiene el valor 'show', los bordes y fondos son dibujados alrededor/detrás de las celdas vacías (como las celdas normales).

Un valor de 'hide' significa que los bordes o fondos no se dibujan alrededor/detrás de las celdas vacías (ver punto 6 en [17.5.1](#)). Mas aun, si todas las celdas de una fila tienen el valor 'hide' y no tienen contenido, la fila entera se comporta como si tuviera 'display: none'.

La siguiente regla provoca que los bordes y fondos sean dibujados alrededor de todas las celdas:

```
table { empty-cells: show }
```

17.6.2 El modelo de bordes cerrados

En el modelo de bordes cerrados, es posible especificar los bordes que rodean a toda o parte de una celda, fila, grupo de filas, columna y grupo de columnas. Los bordes para los atributos "rule" de HTML pueden ser especificados de este modo.

Los bordes son centrados con las líneas de la rejilla entre las celdas. Las aplicaciones del usuario deben encontrar una regla consistente para redondear en caso de un número impar de unidades continuas (píxeles de pantalla, puntos de impresora).

El diagrama de abajo muestra como interactúan el ancho de la tabla, el ancho de los bordes, el relleno, y el ancho de la celda. Su relación está dada por la siguiente ecuación, que se mantiene para cada fila de la tabla:

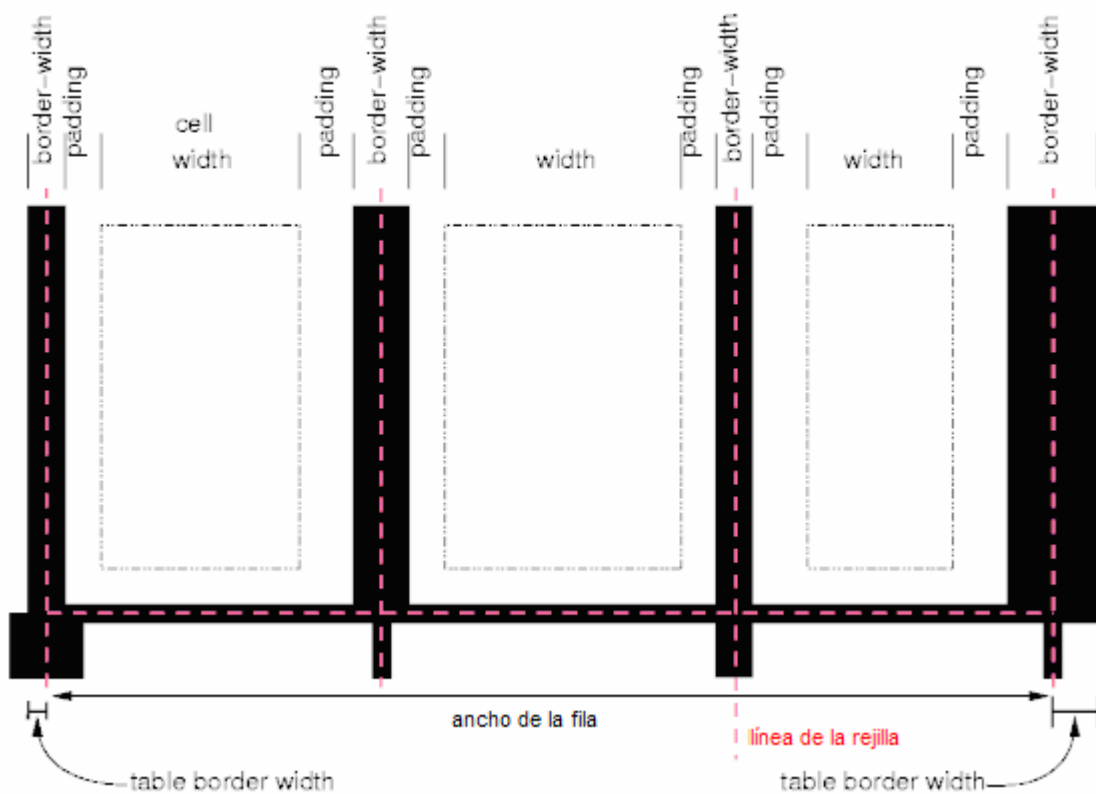
$$\text{row-width} = (0.5 * \text{border-width}_0) + \text{padding-left}_1 + \text{width}_1 + \text{padding-right}_1 + \text{border-width}_1 + \text{padding-left}_2 + \dots + \text{padding-right}_n + (0.5 * \text{border-width}_n)$$

Aquí n es el número de celdas en la fila, padding-left_i y padding-right_i se refieren al relleno izquierdo (respectivamente derecho) de la celda i , y border-width_i se refiere al borde entre las celdas i e $i + 1$.

Las AUs deben calcular un ancho inicial de borde izquierdo y derecho para la tabla mediante el examen de la primer y ultima celda en la primera fila de la tabla. El ancho del borde izquierdo de la tabla es la mitad del borde derecho de la ultima celda cerrada. Si la secuencia de filas tiene muchos bordes derechos e izquierdos cerrados, entonces cualquier exceso se expande dentro del área del margen de la tabla.

El ancho del borde superior de la tabla es calculado examinando todas las celdas cuyos bordes superiores se cierran con el borde superior de la tabla. El ancho del borde de la tabla es igual a la mitad del máximo borde superior cerrado. El borde inferior es calculado examinando todas las celdas cuyo borde inferior se cierra con el inferior de la tabla. El ancho del borde inferior es igual a la mitad del máximo borde inferior cerrado.

Cualquier borde que se extiende dentro del margen se tiene en cuenta cuando se determina si la tabla desborda algún ascendente (ver 'overflow').



[D]

Esquema mostrando el ancho de las celdas y bordes y el relleno de las celdas.

Observe que en este modelo, el ancho de la tabla incluye la mitad del borde de la tabla. Además, en este modelo, una tabla no tiene relleno (pero tiene márgenes).

Resolución de conflictos de bordes

En el modelo de bordes cerrados, los bordes de cada lado de cada celda pueden ser especificados por las propiedades borde sobre una variedad de elementos que se encuentran en ese lado (celdas, filas, grupos de filas, columnas, grupos de columnas, y la propia tabla), y esos bordes pueden variar en grosor, estilo y color. La regla de manejo es que en cada lado el estilo mas "atractivo" de borde es elegido, excepto que cualquier aparición de estilo 'hidden' elimina incondicionalmente el borde.

Las siguientes reglas determinan que estilo "gana" en caso de conflicto:

1. Los bordes con 'border-style' como 'hidden' tiene precedencia sobre todos los demás bordes en conflicto. Cualquier borde con este valor suprime todos los bordes en esa ubicación.
2. Los bordes con un estilo 'none' tienen la prioridad más baja. Solo si las propiedades de borde de todos los elementos que se encuentran en ese lado son 'none' el borde será omitido (pero observe que 'none' es el valor por defecto para el estilo de borde)
3. Si ninguno de los estilos es 'hidden' y al menos uno de ellos no es 'none', entonces los bordes finos son descartados en favor de los mas gruesos. Si varios tienen el mismo 'border-width' entonces los estilos son preferidos en este orden: 'double', 'solid', 'dashed', 'dotted', 'ridge', 'outset', 'groove', y el más bajo: 'inset'.
4. Si los estilos de borde difieren solo en el color, entonces un estilo fijado sobre una celda gana a uno aplicado sobre una fila, la cual gana a un grupo de filas, columnas, grupo de columnas, y finalmente, tabla. If border styles differ only in color, then a style set on a cell wins over one on a row, which wins over a row group, column, column group and, lastly, table. No está definido lo que ocurre cuando dos colores del mismo tipo entran en conflicto.

El siguiente ejemplo ilustra la paliación de estas reglas de precedencia. Esta hoja de estilo:

```
table          { border-collapse: collapse;
                 border: 5px solid yellow; }
*#coll        { border: 3px solid black; }
td             { border: 1px solid red; padding: 1em; }
td.solid-blue  { border: 5px dashed blue; }
td.solid-green { border: 5px solid green; }
```

con este código fuente HTML:

```
<P>
<TABLE>
<COL id="col1"><COL id="col2"><COL id="col3">
<TR id="row1">
  <TD> 1
  <TD> 2
  <TD> 3
</TR>
<TR id="row2">
  <TD> 4
  <TD class="solid-blue"> 5
  <TD class="solid-green"> 6
</TR>
```

```

<TR id="row3">
  <TD> 7
  <TD> 8
  <TD> 9
</TR>
<TR id="row4">
  <TD> 10
  <TD> 11
  <TD> 12
</TR>
<TR id="row5">
  <TD> 13
  <TD> 14
  <TD> 15
</TR>
</TABLE>

```

would produce something like this:

<i>1</i>	<i>2</i>	<i>3</i>
<i>4</i>	<i>5</i>	<i>6</i>
<i>7</i>	<i>8</i>	<i>9</i>
<i>10</i>	<i>11</i>	<i>12</i>
<i>13</i>	<i>14</i>	<i>15</i>

[D]

Un ejemplo de una tabla con bordes cerrados.

Aquí un ejemplo de bordes cerrados invisibles:

foo bar
foo bar

[D]

Tabla con todos bordes internos omitidos.

Código fuente HTML:

```

<TABLE style="border-collapse: collapse; border: solid;">
<TR><TD style="border-right: hidden; border-bottom: hidden">foo</TD>
  <TD style="border: solid">bar</TD></TR>
<TR><TD style="border: none">foo</TD>
  <TD style="border: solid">bar</TD></TR>
</TABLE>

```

17.6.3 Estilos de bordes

Algunos de los valores de '[border-style](#)' tienen diferentes significados para las tablas que para otros elementos. En la lista de abajo están marcados con un asterisco.

none

Ningún borde.

***hidden**

Igual que 'none', pero en el [modelo de bordes cerrados](#), también inhibe cualquier otro borde (ver la sección sobre [conflictos de bordes](#)).

dotted

El borde es una serie de puntos.

dashed

El borde es una serie de segmentos cortos de líneas (línea discontinua).

solid

El borde es un simple segmento de línea.

double

El borde son dos líneas sólidas. La suma de las dos líneas y espacio entre ellas equivalen al valor de '[border-width](#)'.

groove

El borde parece como tallado en el lienzo.

ridge

Lo opuesto de 'groove': el borde parece estar sobresaliendo del lienzo.

***inset**

En el [modelo de bordes separados](#), el borde provoca que toda la caja luzca como si estuviera embebida (incrustada) en el lienzo. En el [modelo de bordes cerrados](#), es igual a 'ridge'.

***outset**

En el [modelo de bordes separados](#), el borde provoca que toda la caja luzca como si estuviera en relieve (sobresaliendo) sobre el lienzo. En el [modelo de bordes cerrados](#), es igual a 'groove'.

18 Interfaz de usuario

Contenido

- [18.1 Cursores: la propiedad 'cursor'](#)
- [18.2 Colores del Sistema CSS2](#)
- [18.3 Preferencias de fuentes del usuario](#)
- [18.4 Contornos dinámicos: la propiedad 'outline'](#)
 - [18.4.1 Contornos y foco](#)
- [18.5 Ampliación](#)

18.1 Cursores: la propiedad 'cursor'

'cursor'

<i>Valor:</i>	[[<u><uri></u> ,]* [auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress]] <u>inherit</u>
<i>Inicial:</i>	auto
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>visual</u> , <u>interactivos</u>
<i>Valor computado:</i>	URI absoluta, sino como el especificado

Esta propiedad especifica el tipo de cursor que se exhibirá para el dispositivo señalizador. Los valores tienen los significados siguientes:

auto

La AU determina el cursor a mostrar basado en el contexto actual.

crosshair

Una cruz simple (ej., segmentos cortos de línea semejantes a un signo "+").

default

El cursor predeterminado dependiente de la plataforma. Procesado a menudo como una flecha.

pointer

El cursor es un puntero que indica un enlace.

move

Indica que algo será movido.

e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize

Indique que algún borde será movido. Por ejemplo, el cursor 'se-resize' se usa cuando el movimiento comienza desde la esquina sudeste de la caja.

text

Indica el texto que puede seleccionarse. Procesado a menudo como barra-l.

wait

Indica que el programa está ocupado y el usuario debe esperar. Procesado a menudo como un reloj de pulsera o un reloj de arena.

progress

Un indicador de progreso. El programa está realizando algún proceso, pero es diferente de 'wait' en el que el usuario todavía puede actuar recíprocamente con el programa. Procesado a menudo como una pelo de playa girando, o una flecha con un reloj o reloj de arena.

help

La ayuda está disponible sobre el objeto señalado por el cursor. Procesado a menudo como un signo de interrogación o un globo.

<uri>

La aplicación del usuario recupera el cursor del recurso señalado por el URI. Si la aplicación del usuario no puede manejar el primer cursor de una lista de cursores, debe procurar manejar el segundo, etc. Si la aplicación del usuario no puede manejar ningún cursor definido por el usuario, debe utilizar el cursor genérico al final de la lista.

```
:link,:visited { cursor: url(example.svg#linkcursor), url(hyper.cur),  
pointer }
```

Este ejemplo fija el cursor en todos los hipervínculos (si está visitado o no) a un cursor SVG externo. Las aplicaciones del usuario que no soporte cursores SVG saltarían simplemente al valor siguiente y procurarían utilizar el cursor "hyper.cur". Si ese formato de cursor no fuera soportado también, la AU saltaría al valor siguiente y procesaría simplemente el cursor 'pointer'.

18.2 Colores Del Sistema CSS2

Nota. *Los colores del sistema CSS2 son desaprobados en el módulo de color de CSS3.*

Además de poder asignar valores de color predefinidos al texto, fondos, etc., CSS2 introduce un sistema de valores de nombres de color que permite a los autores especificar colores de manera que se integren en el entorno gráfico del sistema operativo.

Para los sistemas que no tienen un valor correspondiente, el valor especificado se debe mapear al valor más cercano del sistema, o a un color por defecto.

El siguiente es un listado de los valores adicionales para las propiedades de CSS relacionadas con el color y sus significados generales. Cualquier propiedad de color (ej., 'color' o 'background-color') puede tomar uno de los nombres siguientes. Aunque éstos son sensibles a mayúsculas y minúsculas, se recomienda que la mezcla de mayúsculas y minúsculas mostrada debajo sea utilizada, para hacer los nombres más legibles.

ActiveBorder

Borde de la ventana activa.

ActiveCaption

Título de la ventana activa.

AppWorkspace

Color de fondo de una interfaz de documentos múltiples.

Background

Fondo del escritorio.

ButtonFace

Color de la cara para los elementos tridimensionales de la pantalla.

ButtonHighlight

Color de luz más claro de los elementos tridimensionales de la pantalla (para los bordes mas alejados de la fuente de luz).

ButtonShadow

Color de la sombra de los elementos tridimensionales de la pantalla.

ButtonText

Texto de los botones.

CaptionText

Texto de los títulos, caja de dimensiones y cajas de flecha de las barras de desplazamiento.

GrayText

Texto grisáceo (deshabilitado). Este color se fija a #000 si el manejador de pantalla actual no soporta un color gris sólido.

Hightlight

Elemento(s) (Item(s)) seleccionado(s) en un control.

HighlightText

Texto del(los) elemento(s) (item(s)) seleccionados en un control.

InactiveBorder

Borde de la ventana inactiva.

InactiveCaption

Color del titulo de la ventana inactiva.

InactiveCaptionText

Color del texto en un titulo inactivo.

InfoBackground

Color del fondo para los controles de sugerencias (tooltip).

InfoText

Color del texto para los controles de sugerencias (tooltip).

Menú

Fondo del menú.

MenuText

Texto de los menús.

Scrollbar

Área gris de la barra de desplazamiento.

ThreeDDarkShadow

Sombra oscura de los elementos tridimensionales de pantalla.

ThreeDFace

Color de la cara de los elementos tridimensionales de pantalla.

ThreeDHighlight

Color resaltado de los elementos tridimensionales de pantalla.

ThreeDLightShadow

Color claro de los elementos tridimensionales de pantalla (para los bordes enfrente de la fuente de luz).

ThreeDShadow

Sombra oscura de los elementos tridimensionales de pantalla.

window

Fondo de la ventana.

WindowFrame

Marco de la ventana.

WindowText

Texto de las ventanas.

Por ejemplo, para poner los colores del primero plano y del fondo de un párrafo los mismos colores de primer plano y de fondo de la ventana del usuario, escriba lo siguiente:

```
p { color: WindowText; background-color: Window }
```

18.3 Preferencias de fuentes del usuario

De igual modo que para los colores, los autores pueden especificar las fuentes de manera que hagan uso de los recursos del sistema de un usuario. Consulte por favor la propiedad 'font' para los detalles.

18.4 Contornos dinámicos: la propiedad 'outline'

Ocasionalmente, los autores de hojas de estilo pueden desear crear contornos alrededor de objetos visuales tales como botones, campos activos de formularios, mapas de imágenes, etc., para enfatizarlos. Los contornos de CSS 2.1 se diferencian de los bordes en los siguientes sentidos:

1. Los contornos no ocupan espacio.
2. Los contornos pueden ser no rectangulares.

Las propiedades del contorno controlan el estilo de estos contornos dinámicos.

'outline'

Valor: [<'outline-color'> || <'outline-style'> || <'outline-width'>] | inherit
Inicial: Ver las propiedades individuales
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medio: visual, interactivo
Valor computado: Ver propiedades individuales

'outline-width'

Valor: <border-width> | inherit
Inicial: medium
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medio: visual, interactivos
Valor computado: Medida absoluta; '0' si el estilo outline es 'none' o 'hidden'

'outline-style'

Valor: <border-style> | inherit
Inicial: none
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medio: visual, interactivos
Valor computado: Según el especificado

'outline-color'

Valor:	<color> invert <u>inherit</u>
Inicial:	invert
Se aplica a:	Todos los elementos
Se hereda:	no
Porcentajes:	N/A
Medio:	<u>visual</u> , <u>interactivo</u>
Valor computado:	Según el especificado

El contorno creado con las propiedades de contorno es dibujado "sobre" una caja, es decir, el contorno está siempre encima, y no influye en la posición o el tamaño de la caja, o de ninguna otra caja. Por lo tanto, mostrar u ocultar contornos no causa el redibujado.

El contorno puede ser dibujado comenzando justo fuera del [límite del borde](#).

Los contornos pueden ser no-rectangulares. Por ejemplo, si el elemento está dividido en varias líneas, el contorno es el contorno mínimo que encierra a todas las cajas del elemento. En contraste con los [bordes](#), el contorno no está abierto en la línea de finalización o comienzo de la caja, pero está siempre totalmente conectado si es posible.

La propiedad '[outline-width](#)' acepta los mismos valores que '[border-width](#)'.

La propiedad '[outline-style](#)' acepta los mismos valores que '[border-style](#)', excepto que 'hidden' no sea un estilo lícito de contorno.

La propiedad '[outline-color](#)' acepta todos los colores, como así también la palabra clave 'invert'. Se espera que 'invert' produzca una inversión de color en los píxeles de la pantalla. Esto es un truco común para asegurar que el borde del foco sea visible, sin que importe el color del fondo.

Las AUs con conformidad pueden ignorar el valor 'invert' en las plataformas que no soportan la inversión del color de los píxeles de la pantalla. Si la AU no soporta el valor 'invert' entonces el valor inicial de la propiedad '[outline-color](#)' es el valor de la propiedad '[color](#)', similar al valor inicial de la propiedad '[border-top-color](#)'.

La propiedad '[outline](#)' es una propiedad resumida, y determina las tres '[outline-style](#)', '[outline-width](#)', y '[outline-color](#)'.

Nota. *El contorno es igual en todos los lados. En contraste con los bordes, no existe la propiedad '[outline-top](#)' o '[outline-left](#)'.*

Esta especificación no define cómo se dibujan los contornos solapados, o cómo los contornos se dibujan para las cajas que están parcialmente ocultas detrás de otros elementos.

Nota. *Puesto que el contorno del foco no afecta el formato (es decir, no se deja ningún espacio para él en el modelo de caja), puede tapar otros elementos de la página.*

Aquí hay un ejemplo del dibujo de un contorno grueso alrededor de un elemento `BUTTON`:

```
button { outline-width : thick }
```

Los scripts pueden usarse para cambiar dinámicamente el ancho del contorno, sin provocar el redibujado.

18.4.1 Contornos y foco

Las interfaces gráficas pueden utilizar contornos alrededor de elementos para indicar al usuario qué elemento de la página tiene *el foco*. Estos contornos son sumados a cualquier borde, y pasan entre contornos activados y desactivados sin causar el redibujado del documento. El foco es el sujeto de la interacción del usuario con un documento (ej., introduciendo texto, seleccionando un botón, etc). Las aplicaciones del usuario que dan soporte a los [grupos de medios interactivos](#) deben mantener identificada la ubicación del foco y también debe representar el foco. Esto puede realizarse utilizando contornos dinámicos conjuntamente con la pseudo-clase: `focus`.

Por ejemplo, para dibujar una línea negra gruesa alrededor de un elemento cuando tiene el foco, y una línea roja gruesa cuando es activo, las reglas siguientes pueden ser utilizadas:

```
:focus { outline: thick solid black }  
:active { outline: thick solid red }
```

18.5 Ampliación

El grupo de trabajo de CSS considera que la ampliación de un documento o de partes de un documento no se debe especificar a través de las hojas de estilo. Las aplicaciones del usuario pueden soportar tal ampliación de diversas maneras (ej., imágenes más grandes, sonidos más fuertes, etc.)

Al ampliar una página, las AUs deben preservar las relaciones entre los elementos posicionados. Por ejemplo, una historieta se puede componer de imágenes con elementos de texto superpuestos. Al ampliar esta página, una aplicación del usuario debe mantener el texto dentro del globo de la historieta.

Apéndice A. Hojas de estilo auditivas

Contenido

- [A.1 Tipos de medios 'auditivos' y 'voz'](#)
- [A.2 Introducción a las hojas de estilos auditivas](#)
 - [A.2.1 Ángulos](#)
 - [A.2.2 Tiempos](#)
 - [A.2.3 Frecuencias](#)
- [A.3 Propiedades del volumen: 'volume'](#)
- [A.4 Propiedades del habla: 'speak'](#)
- [A.5 Propiedades de la pausa: 'pause-before', 'pause-after' y 'pause'](#)
- [A.6 Propiedades de la señal: 'cue-before', 'cue-after' y 'cue'](#)
- [A.7 Propiedades de la mezcla: 'play-during'](#)
- [A.8 Propiedades espaciales: 'azimuth' y 'elevation'](#)
- [A.9 Propiedades de las características de la voz: 'speech-rate' 'voz-family' 'pitch' 'pitch-range' 'stress' y 'richness'](#)
- [A.10 Propiedades de lectura: 'speak-punctuation' y 'speak-numeral'](#)
- [A.11 Procesado de tablas de audio](#)
 - [A.11.1 Cabeceras de la voz: la propiedad 'speak-header'](#)
- [A.12 Ejemplos de hojas de estilo para HTML](#)
- [A.13 Emacspeak](#)

Este capítulo es informativo. Las AUs no requieren implementar las propiedades de este capítulo para ser conformes a CSS 2.1.

A.1 Tipos de medios 'auditivos' y 'voz'

Esperamos que en un nivel futuro del CSS haya nuevas propiedades y valores definidos para la salida voz. Por lo tanto CSS 2.1 reserva el tipo de medio de 'voz' (véase el [capítulo 7, "tipos de medios"](#)), pero todavía no lo define que propiedades se aplican o no a él.

Las propiedades en este apéndice se aplican a un tipo de medio 'auditivo', esas fueron introducidas en CSS2. El tipo 'auditivo' ahora se desapueba.

Esto significa que una hoja de estilo tal como

```
@media speech {  
  body { voice-family: Paul }  
}
```

es válido, pero ese significado no es definido por CSS 2.1, mientras que

```
@media auditivos {  
  body { voice-family: Paul }  
}
```

es desaprobado, pero definido por este apéndice.

A.2 Introducción a las hojas de estilo auditivas

El procesamiento sonoro de un documento, comúnmente usado ya por las comunidades de ciegos y discapacitados visuales, combina la síntesis de voz e "íconos sonoros". A menudo tal representación sonora se produce convirtiendo al documento en texto puro y procesándolo con un lector de pantalla --programa o dispositivo que simplemente lee todos los caracteres sobre la pantalla--. Esto resulta en una representación menos efectiva que la que se conseguiría si se mantuviera la estructura del documento. Las hojas de estilo para la presentación sonora pueden utilizarse conjuntamente con las propiedades visuales (mezcla de medios) o como una alternativa sonora a la presentación visual.

Además de las ventajas obvias de la accesibilidad, hay otros mercados grandes para escuchar la información, incluyendo uso en-coches, sistemas de documentación industriales y médicos (intranets), entretenimiento casero, y ayuda a los usuarios que están aprendiendo a leer o que tienen dificultados de lectura.

Al usar propiedades auditivas, el lienzo consiste en un espacio físico tridimensional (sonido circundante) y un espacio temporal (uno puede especificar sonidos antes, durante, y después de otros sonidos). Las propiedades CSS también permiten que los autores varíen la calidad en la sintetización de la voz (tipo de voz, frecuencia, inflexión, etc.).

```
h1, h2, h3, h4, h5, h6 {
    voice-family: paul;
    stress: 20;
    richness: 90;
    cue-before: url("ping.au")
}
p.heidi { azimuth: center-left }
p.peter { azimuth: right }
p.goat { volume: x-soft }
```

Esto ordenará el sintetizador de voz leer los títulos con una voz (una especie de "fuente auditiva") llamado "paul", en un tono llano, pero en voz muy alta. Antes de decir los títulos, se escuchará un archivo de sonido del URL dado. Los párrafos con clase "heidi" parecerán venir del frente-izquierdo (si el sistema de sonidos es soporta el audio espacial), y los párrafos con clase "peter" vendrán de la derecha. Los párrafos con clase "goat" serán muy suaves.

A.2.1 Ángulos

Los valores del ángulo son denotados por <ángulo> en el texto. Su formato es un <número> seguido inmediatamente por un identificador de la unidad del ángulo.

Los identificadores de la unidad del ángulo son:

- **deg**: grados
- **grad**: gradientes
- **rad**: radios

Los valores del ángulo pueden ser negativos. Deben ser normalizados al rango 0-360deg por la aplicación del usuario. Por ejemplo, -10deg y 350deg son equivalentes.

Por ejemplo, un ángulo recto es '90deg' o '100grad' o '1.570796326794897rad'.

Como para <medida>, la unidad puede ser omitida, si el valor es cero: '0deg' se puede escribir como '0'.

A.2.2 Tiempos

Los valores del tiempo se denotan por <time> en el texto. Su formato es un <número> seguido inmediatamente por un identificador de la unidad del tiempo.

Los identificadores de la unidad del tiempo son:

- **ms**: milisegundos
- **s**: segundos

Los valores del tiempo no pueden ser negativos.

Como para <medida>, la unidad puede ser omitida, si el valor es cero: '0s' se puede escribir como '0'.

A.2.3 Frecuencias

Los valores de la frecuencia son denotados por <frequency> en el texto. Su formato es un <número> seguido inmediatamente por un identificador de la unidad de la frecuencia.

Los identificadores de la unidad de la frecuencia son:

- **Hertzio** : Herzio
- **kilociclo** : kilociclos

Los valores de la frecuencia no pueden ser negativos.

Por ejemplo, 200Hz (o 200hz) es un sonido bajo, y 6kHz es un sonido triple.

Como para <medida>, la unidad puede ser omitida, si el valor es cero: '0Hz' se puede escribir como '0'.

A.3 Propiedades del volumen: 'volume'

'volume'

<i>Valor:</i>	<u><número></u> <u><porcentaje></u> silent x-soft soft medium loud x-loud <u>inherit</u>
<i>Inicial:</i>	Médium
<i>Se aplica a:</i>	todos los elementos
<i>Se hereda:</i>	sí
<i>Porcentajes:</i>	Se refiere al valor heredado
<i>Medios:</i>	<u>auditivos</u>
<i>Valor computado:</i>	número

Volume se refiere al volumen medio del sonido. En otras palabras, una voz altamente modulada a un volumen de 50 podría alcanzar bastante más que eso. Los valores globales pueden ser ajustado por las personas para mayor confort, por ejemplo con un control físico del volumen (que aumentaría los valores 0 y 100 proporcionalmente), lo que esta propiedad hace es ajustar el rango dinámico.

Los valores tienen los significados siguientes:

<número>

Cualquier número entre '0' y '100'. '0' representa *el mínimo nivel de volumen audible* y 100 corresponde *al máximo nivel comfortable*.

<porcentaje>

Los valores del porcentaje se calculan con relación al valor heredado, y después se cortan al rango de '0' a '100'.

silent

Ningún sonido en absoluto. El valor '0' no significa lo mismo que 'silent'.

x-soft

Igual a '0'.

soft

Igual a '25'.

medium

Igual a '50'.

loud

Igual a '75'.

x-loud

Igual a '100'.

Las aplicaciones del usuario deben permitir que los valores correspondientes a '0' y '100' sean determinados por el oyente. Ninguna configuración es universalmente aplicable; los valores apropiados dependen del equipamiento en uso (parlantes, auriculares), el medio ambiente (en el automóvil, un equipo hogareño, la biblioteca) y las preferencias personales. Algunos ejemplos:

- Un navegador para usarlo en-coche tiene una configuración apropiada para cuando hay mucho ruido ambiente. '0' equivaldría a un nivel bastante alto y '100' a un nivel realmente alto. La voz es fácilmente audible sobre el ruido del camino pero se reduce el rango dinámico total. Los coches con un aislamiento mejor podrían permitir un rango dinámico más amplia.
- Otro navegador audible se está utilizando en un apartamento, tarde por la noche, o en un cuarto compartido de estudio. '0' se fija a un nivel muy bajo y '100' a un nivel bastante bajo, también. Como en el primer ejemplo, hay un pequeño desnivel; se reduce el rango dinámico. Los volúmenes reales son bajos aquí, mientras que eran altos en el primer ejemplo.
- En casa silenciosa y aislada, un caro equipo casero de alta fidelidad. '0' se fija bastante bajo y '100' es totalmente alto; hay un gran rango dinámico.

La misma hoja de estilo del autor se podía utilizar en todos los casos, simplemente ajustando adecuadamente los puntos '0' y '100' en el lado del cliente.

A.4 Propiedades de voz: 'speak'

'speak'

<i>Valor:</i>	normal none spell-out <u>inherit</u>
<i>Inicial:</i>	normal
<i>Se aplica a:</i>	todos los elementos
<i>Se hereda:</i>	sí
<i>Porcentajes:</i>	N/A
<i>Medios:</i>	<u>auditivos</u>
<i>Valor computado:</i>	Según lo especificado

Esta propiedad especifica si el texto será procesado sonoramente y si es así de qué manera. Los valores posibles son:

none

Suprime el procesamiento sonoro para que el elemento no requiera ningún tiempo para procesarse. Observe, sin embargo, que los descendientes pueden anular este valor y ser expresados verbalmente. (Para asegurarse de suprimir el procesamiento de un elemento y sus descendientes, utilice la propiedad 'display').

normal

Utiliza las reglas de pronunciación adecuadas al idioma para el procesamiento de un elemento y sus descendientes.

Spell-out

Deletrea el texto una letra por vez (útil para las siglas y las abreviaturas).

Observe la diferencia entre un elemento cuya propiedad 'volume' tiene un valor 'silent' y de un elemento cuya propiedad 'speak' tiene el valor 'none'. En el primero el procesamiento toma el mismo tiempo como si hubiera sido hablado, incluyendo cualquier pausa antes y después del elemento, pero no se genera ningún sonido. El segundo no requiere ningún tiempo y no se procesa (aunque sus descendientes pueden serlo).

A.5 Propiedades de la pausa: 'pause-before', 'pause-after', y 'pause'

'pause-before'

<i>Valor:</i>	<u><tiempo></u> <u><porcentaje></u> <u>inherit</u>
<i>Inicial:</i>	0
<i>Se aplica a:</i>	todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	vea el texto
<i>Medios:</i>	<u>auditivos</u>
<i>Valor computado:</i>	tiempo

'pause-after'

Valor: < tiempo > | < porcentaje > | inherit
Inicial: 0
Se aplica a: todos los elementos
Se hereda: no
Porcentajes: vea el texto
Medios: auditivos
Valor computado: tiempo

Estas propiedades especifican una pausa que se observará antes (o después) de leer (el programa para leer textos) el contenido de un elemento. Los valores tienen los significados siguientes:

<tiempo>

Expresa la pausa en unidades absolutas de tiempo (segundos y milisegundos).

<porcentaje>

Se refiere al inverso del valor de la propiedad 'speech-rate'. Por ejemplo, si speech-rate es de 120 palabras por minuto (es decir, una palabra toma medio segundo, o 500ms) entonces un 'pause-before' del 100% significa una pausa de 500ms y un 'pause-before' de 20% significa 100ms.

Las pausas se inserta entre el contenido del elemento y cualquier contenido 'cue-before' o 'cue-after'.

Los autores deben utilizar unidades relativas para crear hojas de estilo más robustas ante los grandes cambios de velocidad de la lectura (speech-rate).

'pause'

Valor: [[<tiempo> | <porcentaje>] {1,2}] | inherit
Inicial: vea las propiedades individuales
Se aplica a: todos los elementos
Se hereda: no
Porcentajes: vea las descripciones de 'paues-before' y 'pause-after'
Medios: auditivos
Valor computado: vea las propiedades individuales

La propiedad 'pause' es una forma abreviada para fijar 'pause-before' y 'pause-after'. Si se dan dos valores, el primer valor es 'pause-before' y el segundo es 'pause-after'. Si se da solamente un valor, se aplica a ambas propiedades.

```
h1 { pause: 20ms } /* pause-before: 20ms; pause-after: 20ms */
h2 { pause: 30ms 40ms } /* pause-before: 30ms; pause-after: 40ms */
h3 { pause-after: 10ms } /* pause-before unspecified; pause-after: 10ms */
```

A.6 Propiedades de la señal: 'cue-before', 'cue-after', y 'cue'

'cue-before'

Valor: < uri > |none|inherit
Inicial: None
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medios: auditivos
*Valor computado:*URI absoluto o 'none'

'cue-after'

Valor: < uri > |none|inherit
Inicial: None
Se aplica a: Todos los elementos
Se hereda: no
Porcentajes: N/A
Medios: auditivos
*Valor computado:*URI absoluto o 'none'

Los iconos sonoros son otra manera de distinguir elementos semánticos. Los sonidos se pueden oír antes y/o después del elemento para delimitarlo. Los valores tienen los significados siguientes:

< uri >

El URI debe señalar un recurso de icono sonoro. Si el URI resuelve algo distinto de un archivo de audio, tal como una imagen, el recurso debe ser ignorado y se trata la propiedad como si tuviera el valor 'none'.

none

No se especifica ningún icono sonoro.

```
a {cue-before: url("bell.aiff"); cue-after: url("dong.wav") }  
h1 {cue-before: url("pop.au"); cue-after: url("pop.au") }
```

'cue'

Valor: [< 'cue-before' > || < 'cue-after' >]|inherit
Inicial: vea las propiedades individuales
Se aplica a: todos los elementos
Se hereda: no
Porcentajes: N/A
Medios: auditivos
*Valor computado:*vea las propiedades individuales

La propiedad 'cue' es una forma resumida para fijar 'cue-before' y 'cue-after'. Si se especifican dos valores, el primer valor es 'cue-before' y el segundo es 'cue-after'. Si se especifica solamente un valor, se aplica a ambas propiedades.

Las siguientes dos reglas son equivalentes:

```
h1 {cue-before: url("pop.au"); cue-after: url("pop.au") }  
h1 {cue: url("pop.au") }
```

si una aplicación del usuario no puede procesar un icono sonoro (ej., el sistema del usuario no lo permite), recomendamos que produzca una señal alternativa.

Por favor vea la sección sobre [los pseudo-elementos :before y :after](#) para información acerca de otras técnicas de generación de contenidos. Los sonidos 'cue-before' y los huecos 'pause-before' son insertados antes del contenido del pseudo elemento ':before'. Así mismo, los huecos 'pause-after' y los sonidos 'cue-after' son insertados después del contenido del pseudo elemento ':after'.

A.7 Propiedades de mezcla: 'play-during'

'play-during'

<i>Valor:</i>	<u><uri></u> [mix repeat]? auto none <u>inherit</u>
<i>Inicial:</i>	auto
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	no
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>auditivos</u>
<i>Valor computado:</i>	URI absoluta, el resto según el especificado

Similar a las propiedades 'cue-before' y 'cue-after', estas propiedades especifican un sonido que será escuchado como fondo mientras el contenido de un elemento es leído. Los valores tienen los siguientes significados:

<uri>

El sonido designado por este <uri> se escucha como fondo mientras el contenido del elemento es leído.

mix

Cuando está presente, esta palabra clave significa que el sonido heredado de la propiedad 'play-during' del elemento padre continua escuchándose y el sonido designado por el <uri> es mezclado con él. Si 'mix' no es especificado, el sonido de fondo del elemento reemplaza al del padre.

repeat

Cuando está presente, esta palabra clave significa que el sonido se repetirá si es demasiado corto para completar toda la duración del elemento. De otro modo, el sonido se escucha una vez y luego se detiene. Esto es similar a la propiedad 'background-repeat'. Si el sonido es demasiado largo para el elemento, es cortado una vez que el elemento ha sido leído.

auto

El sonido del elemento padre continúa escuchándose (no es reiniciado, el cual sería el caso si la propiedad hubiera sido heredada).

none

Esta palabra clave significa que hay silencio. El sonido del elemento padre (si lo hay) es silenciado durante el elemento actual y continúa después del elemento actual.

```
blockquote.sad { play-during: url("violins.aiff") }
blockquote Q   { play-during: url("harp.wav") mix }
span.quiet    { play-during: none }
```

A.8 Propiedades espaciales: 'azimuth' y 'elevation'

El audio espacial es una importante propiedad de estilo para la presentación auditiva. Proporciona un modo natural de articular varias voces separadamente, como en la vida real (las personas raramente están todas en el mismo lugar en un cuarto). Los altavoces estereo producen sonidos laterales. Los auriculares o los equipos caseros de 5 altavoces de creciente popularidad pueden generar un sonido completamente envolvente, y los equipos de múltiples altavoces pueden crear un verdadero sonido tridimensional. VRML 2.0 también incluye el audio espacial, lo que implica que en poco tiempo el equipamiento de audio espacial aun precio accesible para el consumidor estará disponible más ampliamente.

'azimuth'

<i>Valor:</i>	<u><ángulo></u> [[left-side far-left left center-left center center-right right far-right right-side] behind] leftwards rightwards <u>inherit</u>
<i>Inicial:</i>	center
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>auditivos</u>
<i>Valor computado:</i>	Ángulo normalizado

Los valores tienen los siguientes significados:

<ángulo>

La posición se describe en términos de un ángulo dentro del rango '-360deg' a '360deg'. El valor '0deg' significa delante en el centro del espacio de sonido. '90deg' es a la derecha, '180deg' atrás, y '270deg' (o, de modo equivalente y más conveniente, '-90deg') a la izquierda.

left-side

Igual a '270deg'. Con 'behind', '270deg'.

far-left

Igual a '300deg'. Con 'behind', '240deg'.

left

Igual a '320deg'. Con 'behind', '220deg'.

center-left

Igual a '340deg'. Con 'behind', '200deg'.

center

Igual a '0deg'. Con 'behind', '180deg'.

center-right

Igual a '20deg'. Con 'behind', '160deg'.

right

Igual a '40deg'. Con 'behind', '140deg'.

far-right

Igual a '60deg'. Con 'behind', '120deg'.

right-side

Igual a '90deg'. Con 'behind', '90deg'.

leftwards

Mueve el sonido a la izquierda, con relación al ángulo actual. Más precisamente, resta 20 grados. La aritmética se lleva en módulos de 360 grados. Observe que 'leftwards' es más exacto describirlo como "rotado en sentido contrario a las agujas", debido a que *siempre* resta 20 grados, aun si el azimuth ya está detrás del oyente (en cuyo caso el sonido parece en realidad moverse a la derecha).

rightwards

Mueve el sonido a la derecha, con relación al ángulo actual. Más precisamente, agrega 20 grados. Ver 'leftwards' para la aritmética.

Esta propiedad es más probable que sea implementada mezclando la misma señal en distintos canales a deferente volumen. Podría utilizar también el cambio de fase, pausa digital, y otras técnicas para proporcionar la ilusión de un espacio de sonido. Los medios exactos para lograr este efecto y el número de altavoces utilizados para hacerlo dependen de la aplicación del usuario, esta propiedad solo identifica el resultado final deseado.

```
h1 { azimuth: 30deg }
td.a { azimuth: far-right } /* 60deg */
#12 { azimuth: behind far-right } /* 120deg */
p.comment { azimuth: behind } /* 180deg */
```

Si el azimuth espacial es especificado y el dispositivo de salida no puede producir sonidos *detrás* de la posición del oyente, las aplicaciones del usuario deben convertir los valores en el hemisferio de atrás en valores hacia adelante del hemisferio. Un método es el siguiente:

- si $90\text{deg} < x \leq 180\text{deg}$ then $x := 180\text{deg} - x$
- si $180\text{deg} < x \leq 270\text{deg}$ then $x := 540\text{deg} - x$

'elevation'

Valor: <ángulo> | below | level | above | higher | lower | inherit
Inicial: level
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: ángulo normalizado

Los valores de esta propiedad tienen los siguientes significados:

<ángulo>

Especifica la evolución como un ángulo, entre '-90deg' y '90deg'. '0deg' significa el horizonte de enfrente, que vagamente significa el nivel con el oyente. '90deg' significa directamente sobre la cabeza y '-90deg' significa directamente debajo.

below

Igual a '-90deg'.

level

Igual a '0deg'.

above

Igual a '90deg'.

higher

Agrega 10 grados a la elevación actual.

lower

Resta 10 grados a la elevación actual.

Los medios exactos para lograr este efecto y el número de altavoces utilizados para hacerlo son indefinidos. Esta propiedad solo identifica el resultado final deseado.

```
h1 { elevation: above }
tr.a { elevation: 60deg }
tr.b { elevation: 30deg }
tr.c { elevation: level }
```

A.9 Propiedades de las características de la voz: 'speech-rate', 'voice-family', 'pitch', 'pitch-range', 'stress', y 'richness'

'speech-rate'

Valor: <número> | x-slow | slow | medium | fast | x-fast | faster | slower | inherit
Inicial: medium
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: numero

Esta propiedad especifica la velocidad de lectura. Observe que se permiten tanto palabras clave absolutas como relativas (compare con 'font-size'). Los valores tienen los siguientes significados:

<número>

Especifica la velocidad de lectura en palabras por minuto, una cantidad que varía un poco según el idioma pero no obstante es ampliamente soportada por los sintetizadores de voz.

x-slow

Igual a 80 palabras por minuto.

slow

Igual a 120 palabras por minuto

medium

Igual a 180 - 200 palabras por minuto.

fast

Igual a 300 palabras por minuto.

x-fast

Igual a 500 palabras por minuto.

faster

Agrega 40 palabras por minuto a la velocidad de lectura actual.

slower

Resta 40 palabras por minuto de la velocidad de lectura actual.

'voice-family'

<i>Valor:</i>	<code>[(<voz-específica> <voz-genérica>),]* [<voz-específica> <voz-genérica>] inherit</code>
<i>Inicial:</i>	Depende de la aplicación del usuario
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentaje:</i>	N/A
<i>Medio:</i>	<u>auditivos</u>
<i>Valor computado:</i>	Según el especificado

El valor es una lista de prioridades separadas por coma, de nombres de familias (compare con 'font-family'). Los valores tienen los siguientes significados:

<voz-genérica>

Los valores son familias de voces. Posibles valores son 'male' (masculina), 'female' (femenina), y 'child' (hijo).

<voz-específica>

Los valores son instancias específicas (Ej., comedian, trinoids, carlos, lani).

```
hl { voice-family: announcer, male }
p.part.romeo { voice-family: romeo, male }
p.part.juliet { voice-family: juliet, female }
```

Los nombres de las voces específicas pueden ir entre comilladas, y de hecho deben ir entre comillas si alguna de las palabras que conforman el nombre no están conformes a las reglas sintácticas de los [identificadores](#). también se recomienda entrecomillar las voces específicas con un nombre formado por más de una palabra. Si las comillas son omitidas, cualquier carácter de [espacio en blanco](#) antes y después del nombre de la familia de voces es ignorado y cualquier secuencia de caracteres de espacios en blanco dentro del nombre de la familia de voces se convierte en un espacio simple.

'pitch'

<i>Valor:</i>	<code><frecuencia> x-low low medium high x-high inherit</code>
<i>Inicial:</i>	medium
<i>Se aplica a:</i>	Todos los elementos
<i>Se hereda:</i>	Si
<i>Porcentajes:</i>	N/A
<i>Medio:</i>	<u>auditivos</u>
<i>Valor computado:</i>	frecuencia

Especifica el tono promedio (frecuencia) de la voz articulada. El tono promedio (frecuencia) de una voz depende de la familia de voces. Por ejemplo, el tono promedio para una voz masculina estándar está alrededor de 120Hz, pero para una voz femenina, está alrededor de 210Hz.

Los valores tienen los siguientes significados:

<frecuencia>

Especifica el tono promedio de la voz articulada en hertzios (Hz).

x-low, low, medium, high, x-high

Estos valores no equivalen a un mapa de frecuencias absolutas porque estos valores dependen de la familia de voces. Las aplicaciones del usuario deben distribuir estos valores con frecuencias apropiadas en base a la familia de voces y al entorno del usuario. Sin embargo, las aplicaciones del usuario deben distribuir esos valores en orden (es decir, 'x-low' es una frecuencia más baja que 'low', etc.).

'pitch-range'

Valor: <número> | inherit
Inicial: 50
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computable: Según los especificado

Especifica la variación en el tono promedio. El tono promedio de una voz humana está determinado por la frecuencia básica y típicamente tiene un valor de 120Hz para la voz masculina y 210Hz para la voz femenina. Los idiomas humanos son hablados variando la inflexión y el tono; estas variaciones transmiten significados y énfasis adicionales. Así, una voz muy animada, es decir, una con muchas inflexiones, muestra una amplia gama de tonos. Esta propiedad especifica el rango dentro del cual se producen estas variaciones, es decir, cuanto se puede desviar la frecuencia básica del tono promedio.

Los valores tienen los siguientes significados:

<número>

Un valor entre '0' y '100'. Un rango de tonos de '0' produce voces monótonas, planas. Un rango de tonos de 50 produce inflexiones normales. Los rangos de tonos mayores a 50 producen voces animadas.

'stress'

Valor: <número> | inherit
Inicial: 50
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: Según el especificado

Especifica la altura de los "picos locales" en el perfil de entonación de una voz. Por ejemplo, el inglés es un idioma **acentuado**, y las diferentes partes de una oración tienen asignados énfasis primarios, secundarios y terciarios. El valor 'stress' controla la cantidad de inflexiones que resultan de estos acentos. Esta propiedad es compañera de la propiedad 'pitch-range' y se proporciona para permitir a los diseñadores aprovechar los sistemas de recepción más fieles.

Los valores tienen los siguientes significados:

<número>

Un valor, entre '0' y '100'. El significado de los valores depende del idioma en que se habla. Por ejemplo, un nivel de '50' para una voz en inglés masculino estándar (tono promedio = 122Hz), hablando con entonación y énfasis normales podría tener un significado distinto a '50' para una voz italiana.

'richness'

Valor: <número> | inherit
Inicial: 50
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: Según el especificado

Especifica la riqueza, o claridad, de la voz que habla (lee). Una voz viva será "transportada" por un cuarto grande, una voz suave no. (El término "suave" se refiere a como se ve la forma de la onda cuando se dibuja)

Los valores tienen los siguientes significados:

<número>

Un valor entre '0' y '100'. Mientras más alto es el valor, más será transportada. Un valor más bajo producirá una voz melifluida, suave.

A.10 Propiedades de lectura: 'speak-punctuation' y 'speak-numeral'

Una propiedad de la lectura adicional, 'speak-header', es descrita debajo.

'speak-punctuation'

Valor: code | none | inherit
Inicial: none
Se aplica a: Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: Según el especificado

Esta propiedad especifica como se leen los signos de puntuación. Los valores tienen los siguientes significados:

code

Los signos de puntuación como puntos y comas, llaves y demás se pronuncian (leen) literalmente.

none

Los signos de puntuación no se pronuncian (leen), pero en cambio son procesados como pausas naturales.

'speak-numeral'

Valor: dígito | continuo | inherit
Inicial: continuo
Se aplica a : Todos los elementos
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: Según el especificado

Esta propiedad controla como se expresan los números. Los valores tienen los siguientes significados:

dígito

Se expresan (leen) los números como dígitos individuales. Así, "237" se lee "dos tres siete".

continuo

Se expresan los números como un número completo. Así, "237" se lee "Doscientos treinta y siete ". La representación en palabras depende del idioma.

A.11 Procesamiento auditivo de tablas

Cuando una tabla es leída por un sintetizador de voz, la relación entre los datos de las celdas y los títulos de las celdas deben expresarse de diferente manera según la alineación horizontal y vertical. Algunos navegadores de voz pueden permitir al usuario moverse alrededor de los dos espacios dimensionales, dando así entonces la oportunidad de mapear la representación espacial de relaciones. Cuando eso no sea posible, la hoja de estilo debe especificar que puntos de títulos son leídos.

A.11.1 Leyendo títulos: la propiedad 'speak-header'

'speak-header'

Valor: once | always | inherit
Inicial: once
Se aplica a: Elementos que tienen información de título de tabla
Se hereda: Si
Porcentajes: N/A
Medio: auditivos
Valor computado: Según el especificado

Esta propiedad especifica si los títulos de la tabla son leídos antes de cada celda o si solo antes de una celda cuando esa celda está asociada con un título diferente de la celda anterior. Los valores tienen los siguientes significados:

once

La celda es leída una vez, antes de una serie de celdas.

always

Los títulos son leídos antes de cada celda a la que pertenece.

Cada lenguaje del documento puede tener diferentes mecanismos que permita a los autores especificar los títulos. Por ejemplo, en HTML 4.0 ([HTML40](#)), es posible especificar información de títulos con tres atributos diferentes ("headers", "scope", y "axis"), y la especificación da un algoritmo para determinar la información de título cuando estos atributos no son especificados.

Informe de Gastos de Viaje

	Comidas	Hoteles	Transportes	subtotal
San Jose				
25-Aug-97	37.74	112.00	45.00	
26-Aug-97	27.28	112.00	45.00	
subtotal	65.02	224.00	90.00	379.02
Seattle				
27-Aug-97	96.25	109.00	36.00	
28-Aug-97	35.00	109.00	36.00	
subtotal	131.25	218.00	72.00	421.25
Total	196.27	442.00	162.00	800.27

[D]

La imagen de una tabla con títulos de celdas ("San José" y "Seattle") que no están en la misma columna o fila que los datos a los que se aplican.

El informe de gastos de viaje que este ejemplo HTML presenta el dinero gastado en comidas, hoteles y transporte en dos lugares (San José y Seattle) durante días sucesivos. Conceptualmente, puede pensar en la tabla en términos de n espacio dimensional. Los títulos de este espacio son: lugares, día, categoría y subtotal. Algunas celdas definen marcas a lo largo de un eje mientras que otras dan el puntos dentro de ese espacio como el dinero gastado. La sintaxis para esta tabla es:

```
<TABLE>
<CAPTION>Informe de Gastos de Viaje</CAPTION>
<TR>
  <TH></TH>
  <TH>Comidas</TH>
  <TH>Hoteles</TH>
  <TH>Transportes</TH>
  <TH>subtotal</TH>
</TR>
<TR>
  <TH id="san-jose" axis="san-jose">San Jose</TH>
</TR>
<TR>
  <TH headers="san-jose">25-Aug-97</TH>
  <TD>37.74</TD>
  <TD>112.00</TD>
  <TD>45.00</TD>
  <TD></TD>
</TR>
<TR>
```

```

    <TH headers="san-jose">26-Aug-97</TH>
    <TD>27.28</TD>
    <TD>112.00</TD>
    <TD>45.00</TD>
    <TD></TD>
</TR>
<TR>
    <TH headers="san-jose">subtotal</TH>
    <TD>65.02</TD>
    <TD>224.00</TD>
    <TD>90.00</TD>
    <TD>379.02</TD>
</TR>
<TR>
    <TH id="seattle" axis="seattle">Seattle</TH>
</TR>
<TR>
    <TH headers="seattle">27-Aug-97</TH>
    <TD>96.25</TD>
    <TD>109.00</TD>
    <TD>36.00</TD>
    <TD></TD>
</TR>
<TR>
    <TH headers="seattle">28-Aug-97</TH>
    <TD>35.00</TD>
    <TD>109.00</TD>
    <TD>36.00</TD>
    <TD></TD>
</TR>
<TR>
    <TH headers="seattle">subtotal</TH>
    <TD>131.25</TD>
    <TD>218.00</TD>
    <TD>72.00</TD>
    <TD>421.25</TD>
</TR>
<TR>
    <TH>Total</TH>
    <TD>196.27</TD>
    <TD>442.00</TD>
    <TD>162.00</TD>
    <TD>800.27</TD>
</TR>
</TABLE>

```

Para proporcionar el modelo de datos en este modelo, los autores hacen esto posible para navegadores con lector de voz activado para explorar la tabla del modo correcto, por ejemplo cada celda podría ser leída como una lista, repitiendo los títulos aplicables antes de cada celda de datos:

```

San José, 25-Aug-97, Comidas: 37.74
San José, 25-Aug-97, Hoteles: 112.00
San José, 25-Aug-97, Transportes: 45.00

```

...

El navegador puede también leer los títulos solo cuando cambien:

```
San José, 25-Aug-97, Comidas: 37.74
  Hoteles: 112.00
  Transportes: 45.00
26-Aug-97, Comidas: 27.28
  Hoteles: 112.00
...
```

A.12 Ejemplo de hoja de estilo para HTML

Esta hoja de estilo describe un posible procesamiento de HTML 4.0:

```
@media auditivos {
h1, h2, h3,
h4, h5, h6      { voice-family: paul, male; stress: 20; richness: 90 }
h1              { pitch: x-low; pitch-range: 90 }
h2              { pitch: x-low; pitch-range: 80 }
h3              { pitch: low; pitch-range: 70 }
h4              { pitch: medium; pitch-range: 60 }
h5              { pitch: medium; pitch-range: 50 }
h6              { pitch: medium; pitch-range: 40 }
li, dt, dd      { pitch: medium; richness: 60 }
dt              { stress: 80 }
pre, code, tt   { pitch: medium; pitch-range: 0; stress: 0; richness: 80 }
em              { pitch: medium; pitch-range: 60; stress: 60; richness: 50 }
strong          { pitch: medium; pitch-range: 60; stress: 90; richness: 90 }
dfn             { pitch: high; pitch-range: 60; stress: 60 }
s, strike       { richness: 0 }
i               { pitch: medium; pitch-range: 60; stress: 60; richness: 50 }
b               { pitch: medium; pitch-range: 60; stress: 90; richness: 90 }
u               { richness: 0 }
a:link          { voice-family: harry, male }
a:visited       { voice-family: betty, female }
a:active        { voice-family: betty, female; pitch-range: 80; pitch: x-
high }
}
```

A.13 Emacspeak

Para información, aquí esta una lista de propiedades implementados por Emacspeak, un subsistema de lectura para el editor Emacs.

- voice-family
- stress (pero con un rango diferente de valores)
- richness (pero con un rango diferente de valores)
- pitch (pero con valores diferentes de nombre)
- pitch-range (pero con un rango diferente de valores)

(Agradecemos a T. V. Raman por la información acerca del estado de la implementación para propiedades auditivas)

Apéndice B. Bibliografía

Contenido

- [B.1 Referencias normativas](#)
- [B.2 Referencias informativas](#)

B.1 Referencias normativas

[COLORIMETRY]

"Colorimetry, Second Edition", CIE Publication 15.2-1986, ISBN 3-900-734-00-3.

Disponible en <http://www.hike.te.chiba-u.ac.jp/ikeda/CIE/publ/abst/15-2-86.html>.

[CSS1]

"Cascading Style Sheets, level 1", H. W. Lie and B. Bos, 17 December 1996.

La última versión disponible en <http://www.w3.org/TR/REC-CSS1>

[CSS2]

"Cascading Style Sheets, level 2, CSS2 Specification", B. Bos, H. W. Lie, C. Lilley and I. Jacobs, 12 May 1998,

La última versión disponible en <http://www.w3.org/TR/REC-CSS2>

[FLEX]

"Flex: The Lexical Scanner Generator", Version 2.3.7, ISBN 1882114213.

[HTML40]

"HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 18 December 1997.

La última versión de la especificación está disponible en <http://www.w3.org/TR/REC-html40/>. La recomendación define tres tipos de documentos: Strict, Transitional, y Frameset, todos accesible por la recomendación.

[IANA]

"Assigned Numbers", STD 2, RFC 1700, USC/ISI, J. Reynolds and J. Postel, October 1994.

Disponible en <http://www.ietf.org/rfc/rfc1700.txt>.

[ICC32]

"ICC Profile Format Specification, version 3.2", 1995.

Disponible en <ftp://sgigate.sgi.com/pub/icc/ICC32.pdf>.

[ISO8879]

[ISO 8879:1986](#) "Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)", ISO 8879:1986.

Para la lista de entidades SGML, consultar <ftp://ftp.ifi.uio.no/pub/SGML/ENTITIES/>.

[ISO10646]

"Information Technology - Universal Multiple- Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:1993. La especificación actual también tiene en cuenta las cinco primeras enmiendas de ISO/IEC 10646-1:1993.

Útil [roadmap of the BMP](#) y [roadmap of plane 1](#) documentos que muestran como se asientan los script en cada rango numérico.

[PNG10]

"PNG (Portable Network Graphics) Specification, Version 1.0 specification", T. Boutell ed., 1 October 1996.

Disponible en <http://www.w3.org/pub/WWW/TR/REC-png-multi.html>.

[RFC1808]

"Relative Uniform Resource Locators", R. Fielding, June 1995.
Disponible en <http://www.ietf.org/rfc/rfc1808.txt>.

[RFC2045]

"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996.
Disponible en <http://www.ietf.org/rfc/rfc2045.txt>. Observe que este RFC hace obsoletos RFC1521, RFC1522, y RFC1590.

[RFC2068]

"HTTP Version 1.1 ", R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, and T. Berners-Lee, January 1997.
Disponible en <http://www.ietf.org/rfc/rfc2068.txt>.

[RFC2119]

"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997.
Disponible en <http://www.ietf.org/rfc/rfc2119.txt>.

[RFC2318]

"The text/css Media Type", H. Lie, B. Bos, C. Lilley, March 1998.
Disponible en <http://www.ietf.org/rfc/rfc2318.txt>.

[RFC1738]

"Uniform Resource Locators", T. Berners-Lee, L. Masinter, and M. McCahill, December 1994.
Disponible en <http://www.ietf.org/rfc/rfc1738.txt>.

[RFC2781]

"UTF-16, an encoding of ISO 10646", P. Hoffman, F. Yergeau, February 2000
Disponible en <http://www.ietf.org/rfc/rfc2781.txt>.

[SRGB]

"Proposal for a Standard Color Space for the Internet - sRGB", M. Anderson, R. Motta, S. Chandrasekar, M. Stokes.
Disponible en <http://www.w3.org/Graphics/Color/sRGB.html>.

[UNICODE]

The Unicode Consortium. The Unicode Standard, Version 4.0.0, defined by: The Unicode Standard, Version 4.0 (Boston, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1)
Disponible en <http://www.unicode.org/versions/Unicode4.0.0/>

[URI]

"Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, August 1998.
Disponible en <http://www.ietf.org/rfc/rfc2396.txt>.

[XML10]

"Extensible Markup Language (XML) 1.0 (third edition)" T. Bray, J. Paoli, C.M. Sperberg-McQueen, Eve Maler, François Yergeau, editors, 4 February 2004.
Disponible en <http://www.w3.org/TR/REC-xml>.

[YACC]

"YACC - Yet another compiler compiler", S. C. Johnson, Technical Report, Murray Hill, 1975.

B.2 Referencias informativas

[CHARSETS]

Registered charset values. Download a list of registered charset values from <http://www.iana.org/assignments/character-sets>.

[CSS3SEL]

"Selectors", D. Glazman, T. Çelik, I. Hickson, 13 November 2001
Disponible en <http://www.w3.org/TR/2001/CR-css3-selectors-20011113>

[DOM]

"Document Object Model Specification", L. Wood, A. Le Hors, 9 October 1997.
Disponible en <http://www.w3.org/TR/WD-DOM/>

[MATH20]

"Mathematical Markup Language (MathML) Version 2.0", D. Carlisle, P. Ion, R. Miner, N. Poppelier, 21 February 2001
Disponible en <http://www.w3.org/TR/2001/REC-MathML2-20010221>

[P3P]

"The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, 16 April 2002
Disponible en <http://www.w3.org/TR/2002/REC-P3P-20020416>

[RFC3066]

"Tags for the Identification of Languages", H. Alvestrand, January 2001.
Disponible en <http://www.ietf.org/rfc/rfc3066.txt>.

[SVG10]

"Scalable Vector Graphics (SVG) 1.0 Specification", J. Ferraiolo, 4 September 2001
Disponible en <http://www.w3.org/TR/2001/REC-SVG-20010904>

[WAI-PAGEAUTH]

"Web Content Accessibility Guidelines", W. Chisholm, G. Vanderheiden, I. Jacobs eds.
Disponible en <http://www.w3.org/TR/WD-WAI-PAGEAUTH/>.

[XHTML]

"XHTML 1.0 The Extensible HyperText Markup Language", various authors,
Disponible en <http://www.w3.org/TR/xhtml1/>.

[XMLNAMESPACES]

"Namespaces in XML", T. Bray, D. Hollander, A. Layman,
Disponible en <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

Apéndice C. Cambios

Contenido

- [C.1 Valores de propiedades adicionales](#)
 - [C.1.1 Sección 4.3.5 Colores](#)
 - [C.1.2 Sección 9.2.4 la propiedad de la 'display'](#)
 - [C.1.3 Sección 12.2 la propiedad 'content'](#)
 - [C.1.4 Sección 18.1 Cursores: la propiedad del 'cursor'](#)
 - [C.1.5 Sección 16.6 Espacios en blanco: la propiedad 'white-space'](#)
- [C.2 Cambios](#)
 - [C.2.1 Sección 3.2 Conformidad](#)
 - [C.2.2 Sección 6.1.2 Valores computados](#)
 - [C.2.3 Sección 6.4.3 Calculo de especificidad de un selector](#)
 - [C.2.4 Sección 6.4.4 Precedencia de las indicaciones de presentación fuera de CSS](#)
 - [C.2.5 Capítulo 9 Modelo de formato visual](#)
 - [C.2.6 Sección 10.3.7 Posicionamiento absoluto, elementos no reemplazados](#)
 - [C.2.7 Sección 10.6.4 Posicionamiento absoluto, elementos no reemplazados](#)
 - [C.2.8 Sección 11.1.2 Recorte: la propiedad 'clip'](#)
 - [C.2.9 Sección 14.2.1 Propiedades del fondo](#)
 - [C.2.10 Sección 17.4.1 Posicionamiento y alineamiento del título](#)
 - [C.2.11 Sección 17.5.4 Alineación horizontal en una columna](#)
 - [C.2.12 Sección 17.6 Bordes](#)
 - [C.2.13 Capítulo 12 Contenido generado, numeración automática y listas](#)
 - [C.2.14 Sección 12.2 La propiedad 'content'](#)
 - [C.2.15 Capítulo 13 Medios paginados](#)
 - [C.2.16 Capítulo 15 Fuentes](#)
 - [C.2.17 Capítulo 16 Texto](#)
 - [C.2.18 Apéndice A Hojas de estilos auditivas](#)
 - [C.2.19 Otros](#)
- [C.3 Errores](#)
 - [C.3.1 Propiedades abreviadas](#)
 - [C.3.2 Sección 4.1.1 \(y G2\)](#)
 - [C.3.3 Sección 4.1.3 Caracteres y caso](#)
 - [C.3.4 Sección 4.3 \(problema doble de signos\)](#)
 - [C.3.5 Sección 4.3.2 Medidas](#)
 - [C.3.6 Sección 4.3.6](#)
 - [C.3.7 Sección 5.10 Pseudo-elementos y pseudo-clases](#)
 - [C.3.8 Sección 8.2 Ejemplos de márgenes, rellenos, y bordes](#)
 - [C.3.9 Sección 8.5.2 Color del borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', y 'border-color'](#)
 - [C.3.10 Sección 8.4 Propiedades de relleno](#)
 - [C.3.11 Sección 8.5.3 Estilo del borde](#)
 - [C.3.12 Sección 8.5.4 Propiedades abreviadas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'](#)
 - [C.3.13 Sección 8.5.4 Propiedades abreviadas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'](#)
 - [C.3.14 Sección 9.3.1](#)
 - [C.3.15 Sección 9.3.2](#)

- [C.3.16 Sección 9.4.3](#)
- [C.3.17 Sección 9.7 Relaciones entre 'display', 'position', y 'float'](#)
- [C.3.18 Sección 10.3.2 Elementos en línea, reemplazados \(y 10.3.4, 10.3.6, y 10.3.8\)](#)
- [C.3.19 Sección 10.3.3](#)
- [C.3.20 Sección 10.6.2 Elementos en línea, reemplazados... \(y 10.6.5\)](#)
- [C.3.21 Sección 10.6.3](#)
- [C.3.22 Sección 11.1.1](#)
- [C.3.23 Visibilidad: la propiedad 'visibility'](#)
- [C.3.24 Sección 12.6.2 Listas](#)
- [C.3.25 Sección 15.2.6](#)
- [C.3.26 Sección 15.5](#)
- [C.3.27 Sección 16.6 Espacios en blanco: la propiedad 'white-space'](#)
- [C.3.28 Sección 17.2 El modelo de la tabla de CSS](#)
- [C.3.29 Sección 17.2.1 Objetos anónimos de la tabla](#)
- [C.3.30 Sección 17.5 Composición visual del contenido de las tablas](#)
- [C.3.31 Sección 17.5 Composición visual del contenido de las tablas](#)
- [C.3.32 Sección 17.5.1 Capas y transparencia de la tabla](#)
- [C.3.33 Sección 17.6.1 El modelo de bordes separados](#)
- [C.3.34 Apéndice D.2 Escáner léxico](#)
- [C.4 Aclaraciones](#)
 - [C.4.1 Sección 2.2 Breve tutorial de CSS2 para XML](#)
 - [C.4.2 Sección 4.1.1](#)
 - [C.4.3 sección 5.5](#)
 - [C.4.4 Sección 5.9 Selectores de la ID](#)
 - [C.4.5 Sección 5.12.1 Pseudo-elemento :first-line](#)
 - [C.4.6 Sección 6.2.1](#)
 - [C.4.7 Sección 6.4 Cascada](#)
 - [C.4.8 Sección 6.4.3 Calculo de especificidad de un selector](#)
 - [C.4.9 Sección 7.3 Tipos de medios reconocidos](#)
 - [C.4.10 Sección 8.1](#)
 - [C.4.11 Sección 8.3.1](#)
 - [C.4.12 Sección 9.4.2](#)
 - [C.4.13 Sección 9.4.3](#)
 - [C.4.14 Sección 9.10](#)
 - [C.4.15 Block-level, elementos no reemplazados en flujo normal](#)
 - [C.4.16 Sección 10.5 Altura del contenido: la propiedad 'height'](#)
 - [C.4.17 Sección 10.8.1](#)
 - [C.4.18 Sección 11.1](#)
 - [C.4.19 Sección 11.1.1](#)
 - [C.4.20 Sección 11.1.2](#)
 - [C.4.21 Sección 12.1 Pseudo-elementos :before y :after](#)
 - [C.4.22 Sección 12.3.2 Inserción comillas con la propiedad 'content'](#)
 - [C.4.23 Sección 12.6.2 Listas](#)
 - [C.4.24 Sección 14.2.2 El fondo](#)
 - [C.4.25 Sección 14.2.1 Propiedades del fondo](#)
 - [C.4.26 Sección 16.1](#)
 - [C.4.27 Sección 16.2 Alineación: la propiedad 'text-align'](#)
 - [C.4.28 sección 17.5.1 Capas y transparencia de la tabla](#)
 - [C.4.29 Sección 17.5.2 Algoritmos de la anchura de la tabla](#)

- o [C.4.30 Sección 17.6.1 El modelo de borders separados](#)
- o [C.4.31 Fronteras alrededor de las celdas vacías: propiedad 'empty-cells'](#)
- o [C.4.32 Sección 17.6.2 El modelo de bordes cerrados](#)
- o [C.4.33 Sección 18.2](#)
- o [C.4.34 sección A.3](#)
- o [C.4.35 Apéndice G.2 Escaner léxico](#)
- o [C.4.36 Apéndice E. Referencias](#)

Este apéndice es informativo, no normativo.

CSS 2.1 es una versión actualizada de CSS2. Los cambios entre la especificación CSS2 (véase [\[CSS2\]](#)) y esta especificación recaen en cinco grupos: errores conocidos, errores tipográficos, clarificaciones, cambios y adicionales. Los errores tipográficos no se enumeran aquí.

C.1 Valores de propiedades adicionales

C.1.1 Sección 4.3.6 Colores

Nuevo valor de color: 'orange' (naranja)

C.1.2 Sección 9.2.4 La propiedad 'display'

Nuevo valor de 'display': 'inline-block'

C.1.3 Sección 12.2 La propiedad 'content'

Nuevo valor de 'content' 'normal'. (porque ':before { content: "" }' genera una caja, mientras que ':before { content: normal }' no. Se llama 'normal' y no 'none', porque 'content: normal' significará otras cosas para otras clases de elementos en CSS3.)

C.1.4 Sección 18.1 Cursores: la propiedad 'cursor'

Nuevo valor de 'cursor': 'progress'

C.1.5 Sección 16.6 Espacios en blanco: la propiedad 'white-space'

Nuevos valores de 'white-space': 'pre-wrap' y 'pre-line'

C.2 Cambios

C.2.1 Sección 3.2 Conformidad

El soporte para las hojas de estilo del usuario ahora es requerido (en la mayoría de los casos), mas que recomendarse.

C.2.2 Sección 6.1.2 Valores computados

El valor computado de una propiedad puede ahora también ser un porcentaje. En particular, las propiedades siguientes ahora heredan el porcentaje si el valor especificado es un porcentaje:

- background-position
- bottom, left, right, top
- height, width
- margin-bottom, margin-left, margin-right, margin-top,
- min-height, min-width
- padding-bottom, padding-left, padding-right, padding-top
- text-indent

Observe que 'text-indent' hereda por defecto, los otros heredan solamente si la palabra clave 'inherit' se especifica.

C.2.3 Sección 6.4.3 Cálculo de la especificidad de un selector

El atributo "style" ahora tiene una especificidad más alta que cualquier regla del estilo.

C.2.4 Sección 6.4.4 Precedencia de las indicaciones de presentación fuera de CSS

Las "presentaciones fuera de CSS" no existen, a excepción de un pequeño atributo fijado en HTML.

C.2.5 Capítulo 9 Modelo del formato de la representación visual

El valor 'compact' para 'display' no existe en CSS 2.1.

C.2.6 Sección 10.3.7 Posicionamiento absoluto, elementos no reemplazados

Los elementos posicionados absolutamente ahora "reducen" su contenido:

Cuando 'height' y 'right' (o 'width' y 'left') son 'auto', la anchura computada del elemento es la anchura del contenido (que usa un algoritmo similar al de las celdas de la tabla) y entonces 'right' (o 'left') se soluciona. CSS2 dijo incorrectamente que 'right' (o 'left') fuese fijado a 0 en ese caso, y entonces la anchura fue resuelta.

C.2.7 Sección 10.6.4 Posicionamiento absoluto, elementos no reemplazados

Los elementos a nivel de bloque fluyen normalmente, los elementos posicionados absolutamente por defecto toman la altura de su contenido ("reducen"). Si 'height' y 'bottom' son ambos 'auto', el valor computado de 'height' se fija a la altura del contenido y entonces 'bottom' se resuelve. CSS2 dijo incorrectamente lo contrario: 'bottom' fue fijado a 0 y entonces la altura fue resuelta.

C.2.8 Sección 11.1.2 Recorte: la propiedad 'clip'

Mientras que CSS2 especificó que los valores de "rect()" compensan los lados respectivos de la caja, las implementaciones actuales interpretan los valores con respecto a los bordes superiores e izquierdos para los cuatro valores (superior, derecho, inferior, e izquierdo). Ésta es ahora la interpretación correcta.

C.2.9 Sección 14.2.1 Propiedades del fondo

Bajo 'background-position', la oración "las palabras claves no se pueden combinar con valores de porcentaje o valores de medida (todas las combinaciones posibles se dan arriba)" es eliminado. Es decir, un valor: 'top 25%' ahora se permite.

C.2.10 Sección 17.4.1 Posicionamiento y alineación del título

Los valores 'left' y 'right' en 'caption-side' se han quitado.

C.2.11 Sección 17.5.4 Alineación horizontal en una columna

El valor de <string> (cadena) para 'text-align' no es parte de CSS 2.1.

C.2.12 Sección 17.6 Bordos

Varios navegadores populares asumen que un valor inicial para 'border-collapse' de 'separate' más bien que 'collapse' o exhibe el comportamiento que está cerca de ese valor, incluso si no implementan realmente el modelo de la tabla de CSS. 'separate' es ahora el valor inicial.

C.2.13 Capítulo 12 Contenido generando, numeración automática, y listas

El valor de 'marker' para 'display' no existe en CSS 2.1

C.2.14 Sección 12.2 La propiedad 'content'

El valor '<uri>' no se utiliza.

C.2.15 Capítulo 13 Medios paginados

Las propiedades 'size', 'marks' y 'page' no están en CSS 2.1.

C.2.16 Capítulo 15 Fuentes

Las propiedades 'font-stretch' y 'font-size-adjust' no existen en CSS 2.1.

Los descriptores de la fuente y la declaración '@font-face' no existen en CSS 2.1.

C.2.17 Capítulo 16 Texto

La propiedad 'text-shadow' no está en CSS 2.1.

C.2.18 Apéndice A Hojas de estilo auditivas

El capítulo 19 sobre las hojas de estilo auditivas ha pasado al apéndice A y no es normativo en CSS 2.1. Las unidades relacionadas (grado, gradiente, radio, ms, s, hertzio, kHz) también se mueven a este apéndice, al igual que la propiedad 'speak-header' desde el capítulo "tablas". Se desaprueba el tipo de medios 'auditivo'. (Un nuevo tipo de medios de 'lectura' se esperan en CSS3.)

C.2.19 otro

El antiguo apéndice C informativo, "Implementación y notas de desempeño de las fuentes" se deja fuera de CSS 2.1.

C.3 Errores

C.3.1 Propiedades abreviadas

Las propiedades abreviadas toman una lista de los valores de las subpropiedades o el valor 'inherit' (heredado). Uno no puede mezclar 'inherit' con otros valores de las subpropiedades pues no sería posible especificar la subpropiedad a la cual 'inherit' es aplicado. Las definiciones de un número de propiedades abreviadas no cumplen esta regla: 'border-top', 'border-right', 'border-bottom', 'border-left', 'border', 'background', 'font', 'list-style', 'cue', y 'outline'.

C.3.2 Sección 4.1.1 (y G2)

- El símbolo "nmchar" también permite el rango "A-z".
- En la regla para "any" (en la sintaxis base), cambie "FUNCTION" por "FUNCTION any*")".

El carácter de guión bajo ("_") se permite en identificadores. Las definiciones de las macros léxicas "nmstart" y "nmchar" ahora lo incluyen.

C.3.3 Sección 4.1.3 Caracteres y mayúsculas/minúsculas

En la tercera parte, agregada al punto 1:

1. con un espacio (u otro carácter de espacio en blanco): "\26 B" ("&B")

el texto siguiente: "en este caso, las aplicaciones del usuario debe tratar un "CR/LF" par (13/10) como un simple carácter de espacio en blanco."

El guión bajo se permite en identificadores. Cambiado "en CSS2, los identificadores [...] puede contener solamente los caracteres [A-Za-z0-9] y los caracteres 161 de la ISO 10646 y superiores, más el guión (-)" a:

En CSS2, los identificadores [...] contienen solamente los caracteres [A-Za-z0-9] y los caracteres 161 de la ISO 10646 y superiores, más el guión (-) y la guión bajo (_)

C.3.4 Sección 4.3 (problema doble de signos)

Varios valores descritos en subsecciones de esta sección permitieron incorrectamente dos signos "+" o "-" en sus comienzos.

C.3.5 Sección 4.3.2 Medidas

El píxel sugerido de referencia se basa en dispositivo de **96 dpi**, no de 90dpi. El ángulo visual es así sobre **0,0213 grados** en vez de 0,0227, y un píxel en la longitud del brazo es cerca de **0,26 milímetros** en vez de 0,28

C.3.6 Sección 4.3.6

Se suprime los comentarios sobre la restricción del rango después de los ejemplos siguientes:

```
em { color: rgb(255,0,0) }  
em { color: rgb(100%, 0%, 0%) }
```

C.3.7 Sección 5.10 Pseudo-elementos y pseudo-clases

En la segunda parte, la oración siguiente era incompleta: "la excepción es ':first-child ', que se puede deducir de la estructura del documento." La pseudo-clase ':lang()' se puede deducir del documento en algunos casos.

C.3.8 Sección 8.2 Ejemplos de márgenes, relleno, y bordes

Los colores en el HTML del ejemplo no emparejaron los colores en la imagen.

C.3.9 Sección 8.5.2 Color del borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', y 'border-color'

El valor 'transparent' también se permite en 'border-top-color', 'border-right-color', etc. Se ha cambiado el valor de la línea ": <color>|inherit" a

Valor: <color>|transparent|inherit

C.3.10 Sección 8.4 Propiedades de relleno

Las cinco propiedades relacionadas con el relleno ('padding', 'padding-top', 'padding-right', 'padding-bottom', y 'padding-left') ahora dicen que no se aplican a las filas de la tabla, a los grupos de fila, a los grupos título, a los grupos de pie, a las columnas, y a los grupos de la columna.

C.3.11 Sección 8.5.3 Estilo de borde

Se cambia la oración "el color de los bordes dibujados para los valores 'groove', 'ridge', 'inset', y 'outset' depende de la propiedad 'color' del elemento" a

El color de los bordes dibujados para los valores 'groove', 'ridge', 'inset', y 'outset' se debe basar en de la propiedad 'border-color' del elemento, pero las AUs pueden elegir su propio algoritmo para calcular los colores reales utilizados. Por ejemplo, si 'border-color' tiene el valor 'silver', entonces una AU podría utilizar un gradiente de colores de blanco a gris oscuro para indicar un borde inclinado.

C.3.12 Sección 8.5.4 Propiedades de borde abreviadas: 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'

Se cambia <'border-top-width'> por <border-width> como la primera opción para el valor de 'border-top', 'border-right', 'border-bottom', y 'border-left', y se cambia <'border-style'> a <border-style>. Para 'border', se cambia <'border-width'> a <border-width> y <'border-style'> a <border-style>.

El valor 'transparent' es también permitido sobre 'border-top', 'border-bottom', 'border-right', 'border-left', y 'border'.

Se cambian las 2 líneas "Valor: [<'border-top-width'> || <'border-style'> || <color> | inherit" a

Valor: [<border-top-width> || <border-style> || [<color> | transparent] | inherit

C.3.13 Sección 8.5.4 Propiedades abreviadas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left, y 'border'

Se cambia <'border-top-width'> a <border-width> como primera opción del valor para 'border-top', 'border-right', 'border-bottom', y 'border-left', y cambia <'border-style'> a <border-style>. Para 'border', cambia <'border-width'> a <border-width> y <'border-style'> a <border-style>.

C.3.14 Sección 9.3.1

La definición del valor 'static' ahora se dice que no se aplica a las propiedades 'top', 'right', 'bottom', y 'left'.

C.3.15 Sección 9.3.2

Las propiedades 'top', 'right', 'bottom', y 'left', incorrectamente referenciadas a las compensaciones con respecto a los límites del contenido de una caja. El límite apropiado es el límite del margen. Así, para 'top', la descripción se lee ahora: "Esta propiedad especifica cuán lejos se desplaza el límite (borde) del margen superior del bloque de contención de la caja."

C.3.16 Sección 9.4.3

En la primera frase, se añade al final de " Una vez que la caja es situada de acuerdo al flujo normal" las palabras "o flotante,".

C.3.17 Sección 9.7 Relaciones entre 'display', 'position', y 'float'

Si un elemento flotante, la propiedad 'display' es fijada como valor a un nivel de bloque, pero no necesariamente 'block'. En el apartado 3, se cambia " De otro modo, si 'float' tiene un valor distinto a 'none', 'display' es fijada a 'block' y la caja es flotante" a una tabla con los valores computados apropiados.

C.3.18 Sección 10.3.2 En línea, elementos reemplazados (y 10.3.4, 10.3.6, y 10.3.8)

Se Cambia:

Un valor especificado de 'auto' para 'width' da la anchura intrínseca del elemento como valor computado.

a:

Si 'width' tiene un valor especificado de 'auto' y 'height' tiene un valor especificado de 'auto', la anchura intrínseca del elemento es el valor computado de 'width'. Si 'width' tiene especificado un valor de 'auto' y 'height' tiene otro valor especificado, entonces el valor computado de 'width' es (anchura intrínseca) * (altura computada) / (altura intrínseca)).

C.3.19 Sección 10.3.3

En la última frase del párrafo siguiente a la ecuación ("Si el valor de 'direction' es 'ltr', esto en cambio pasa a 'margin-left'") sustituyendo 'rtl' por 'ltr'.

C.3.20 Sección 10.6.2 En línea, elementos reemplazados ... (y 10.6.5)

Se cambia:

Si 'height' es 'auto', el valor computado es la altura intrínseca.

a:

Si 'height' tiene especificado un valor de 'auto' y 'width' también lo tiene especificado un valor de 'auto', La altura intrínseca del elemento es el valor computado de 'height'. Si 'height' tiene especificado un valor de 'auto' y 'width' tiene cualquier otro valor especificado, entonces el valor computado de 'height' es $(\text{altura intrínseca}) * (\text{ancho computado}) / (\text{ancho intrínseco})$.

C.3.21 Sección 10.6.3

La altura calculada para elementos a nivel de bloque, no reemplazados en flujo normal, y elementos flotantes, no reemplazados no eran exactamente correctos. Ahora se tiene en cuenta el caso cuando los márgenes no se cierran, debido a la presencia de un relleno o borde.

C.3.22 Sección 11.1.1

El ejemplo de un elemento DIV que contiene un BLOCKQUOTE que contiene otro DIV no fue procesado correctamente. El primer estilo se aplica a ambos DIVs, tan solo la segunda caja DIV debería ser procesada con un borde rojo. El segundo DIV ahora ha sido cambiado a CITE, el cual no tiene un borde rojo.

C.3.23 Sección 11.2 Visibilidad: la propiedad 'visibility'

Se cambia "Inicial" y "Se hereda" a:

Inicial: visible

Se hereda: si

Esto tiene el mismo efecto que la definición original, pero elimina los estados indefinidos del elemento raíz (el cual tiene un problema para las implementaciones DOM).

C.3.24 Sección 12.5.1 Listas

Bajo la propiedad 'list-style', el ejemplo:

```
ul > ul { list-style: circle outside } /* cualquier UL hijo de un UL */
```

nunca podría emparejar una etiqueta válida de HTML (ya que un elemento UL no puede ser un hijo de otro elemento UL). Un LI ha sido insertado en medio.

C.3.25 Sección 15

'Totum' y 'Kodic' no son un 'serif' pero si 'sans-serif'. 'pathang' no es un 'sans-serif' pero si 'serif'.

C.3.26 Sección 15.5

En el punto 2, se cambia "la AU utiliza el descriptor 'font-family'" a "la AU utiliza la propiedad 'font-family'".

En el punto 6, se cambia "pasos 3, 4 y 5" a "pasos 2, 3, 4 y 5".

C.3.27 Sección 16.6 Espacios en blanco: la propiedad 'white-space'

La propiedad 'white-space' se aplica a *todos los elementos*, no solamente a los elementos a nivel de bloque.

C.3.28 Sección 17.2 El modelo de tablas CSS

En la definición de **table-header-group (grupo de títulos de tabla)**, se cambia "pie" a "encabezado" en "las aplicaciones de usuario que imprimen pueden repetir las filas del pie en cada pagina ocupada por una tabla."

C.3.29 Sección 17.2.1 Objetos anónimos de la tabla

Se desplaza el primer punto del texto después de las reglas de generación de listas y se añaden reglas perdidas.

C.3.30 Sección 17.5 Composición visual del contenido de las tablas

La siguiente nota:

Nota. Las celdas de las tablas pueden ser posicionadas relativa y absolutamente, pero esto no es recomendado: el posicionamiento y flotantes quitan una caja del flujo, afectando la alineación de la tabla.

Ha sido corregido como sigue:

Nota. Las celdas de las tablas pueden ser posicionadas, pero esto no es recomendable: el posicionamiento absoluto y fijo, así como flotante, quitan una caja del flujo, afectando al tamaño de la tabla.

C.3.31 Sección 17.5 Composición visual del contenido de las tablas

Se cambia:

Como otros elementos del lenguaje del documento, los elementos internos de la tabla genera cajas rectangulares con contenido, relleno, y bordes. Estos no tienen márgenes, sin embargo.

a:

Como otros elementos del lenguaje del documento, los elementos internos de la tabla genera cajas rectangulares con contenido y bordes. Las celdas tienen relleno también. Los elementos internos de la tabla no tienen márgenes.

C.3.32 Sección 17.5.1 Capas y transparencia en la tabla

Las filas y columnas solo cubren la tabla completa en el modelo de bordes cerrados, no en el modelo de bordes separados. Los puntos 2,3,4 y 5 han sido corregidos para definir el área cubierto por las filas, columnas, grupos de filas y grupos de columnas en los términos en que las celdas lo cubren.

C.3.33 Sección 17.6.1 El modelo de bordes separados

En la imagen, se cambia, "espacio entre celdas" a "espacio entre bordes".

C.3.34 Apéndice G.2 Escaner del léxico

El carácter guión bajo ("_") es permitido en los identificadores. La definición del léxico de los macros "nmstart" y "nmchar" han sido fijados.

Observe que te los comandos son sensibles a mayúsculas y minúsculas, así las mayúsculas A-Z son aceptadas también.

(Los mismo cambios en la Sección 4.1.1, ver [abajo](#).)

C.4 Aclaraciones

C.4.1 Sección 2.2 Breve tutorial de CSS2 para XML

La especificación para las [hojas de estilo PI de XML](#) fueron escritas después de que CSS2 fuese terminado. La primera línea del ejemplo completo de XML no debería haber sido

```
<?XML:stylesheet type="text/css" href="bach.css"?>, pero
```

```
<?xml-stylesheet type="text/css" href="bach.css"?>
```

C.4.2 Sección 4.1.1

DELIM no debería haber incluido las comillas simples o dobles. Referido también a la Sección 4.1.6 sobre cadenas, las cuales deben tener emparejadas comillas dobles o simples a su alrededor.

C.4.3 Sección 5.5

Cerca del final de la Sección, el texto 'Observe que el espacio en blanco sobre cada lado de "" era confuso. La nota no implicaba que el es requerido un espacio en blanco a ambos lados de "" (ya que la gramática no lo requiere en este caso) pero uno puede utilizar espacios en blanco en este caso.

C.4.4 Sección 5.9 Selectores ID

La palabra "precedencia" en el penúltimo párrafo debería haber sido "especificidad"

C.4.5 Sección 5.12.1 pseudo-elemento :first-line

Se añaden algunas aclaraciones al final de la Sección sobre la secuencia de la etiqueta ficticia en el caso de elementos a nivel de bloque.

C.4.6 Sección 6.2.1

El valor 'inherit' causa que el valor de las propiedades sea heredado. Esto se aplica hasta en propiedades para las cuales el valor no se hereda.

C.4.7 Sección 6.4 Cascada

Se cambia "Las reglas especificadas en una hoja de estilo dada anulan las reglas importadas de otras hojas de estilo." a "Las reglas especificadas en una hoja de estilo dada anulan las reglas del mismo peso importadas de otras hojas de estilo."

C.4.8 Sección 6.4.3 Cálculo de especificidad de un selector

Se añade una nota:

La especificidad está basada solamente sobre la forma del selector. En particular, un selector de la forma "[id=p33]" es contado como un atributo de selector (a=0, b=1, c=0), incluso si el atributo `id` es definido como un "ID" en el código DTD del documento.

C.4.9 Sección 7.3 Tipos de medios reconocidos

Se ha añadido un texto para aclarar que los tipos de medios son mutuamente excluyentes.

C.4.10 Sección 8.1

- Para la definición de "padding edge" (Límite del relleno), se borra la frase "El límite del relleno de una caja define el límite del bloque de contención establecido para una caja." Para información sobre bloques de contención, consulte la [Sección 10.1](#).
- El fondo del borde no es especificado por la propiedad `borde`. Se cambio el ultimo párrafo de 8.1 a:

El estilo del fondo del contenido, relleno y área del borde de una caja es especificado por la propiedad '`background`' del elemento generado. Los márgenes del fondo son siempre transparentes.

C.4.11 Sección 8.3.1

Se añade esta nota aclaratoria en el primer punto de la explicación del margen vertical cerrado:

Nota. Las cajas adyacentes pueden ser generadas por elementos que no son relacionados como hermanos o ascendentes.

C.4.12 Sección 9.4.2

La declaración " Cuando una caja en línea es dividida, los márgenes, bordes y rellenos no tienen ningún efecto visual donde se produce la división." ha sido generalizado. Los márgenes, bordes y rellenos no tienen efecto visual cuando una o mas divisiones ocurren.

C.4.13 Sección 9.4.3

El posicionamiento relativo de cajas no siempre establece un nuevo bloque de contención. Se ha cambiado el segundo párrafo en consecuencia.

Se añade un texto aclaratorio y un ejemplo sobre las propiedades 'left', 'right', 'top' y 'bottom' para el posicionamiento relativo.

C.4.14 Sección 9.10

En esta oración el párrafo ultimo:

Conforme a HTML las aplicaciones de usuarios pueden por lo tanto ignorar las propiedades 'direction' y 'unicode-bidi' en las hojas de estilo del autor y usuario.

La palabra "ignorar" significa que si un valor 'unicode-bidi' o 'direction' tiene conflictos con el valor del atributo "dir" de HTML 4.0, entonces la aplicación del usuario puede elegir utilizar el valor de "dir" en lugar de las propiedades CSS.

Las aplicaciones del usuario no necesitan soportar las propiedades 'direction' y 'unicode-bidi' para conformarse con CSS2 a no ser que estas soporten el procesamiento de texto bi-direccional (excepto para el caso de HTML 4.0 como se observar arriba).

La oración ha sido rescrita para ser más clara.

C.4.15 Sección 10.3.3 Elementos a nivel de bloque, no reemplazados en flujo normal

Se añade la siguiente nota al final de la Sección:

Observe que 'width' no debe ser mayor que 'max-width' y menor que 'min-width'. En particular, no debe ser negativo. Vea las reglas en la sección 10.4 debajo.

C.4.16 Sección 10.5 Altura del contenido: la propiedad 'height'

La AU es libre de elegir el bloque de contención para el elemento raíz (ver 10.1), por lo tanto esta oración ha sido agregada como una sugerencia.:

Una AU puede calcular una altura en porcentaje del elemento raíz relativa al acceso visual (viewport).

C.4.17 Sección 10.8.1

Aclaración de este párrafo:

Observe que los elementos reemplazados tienen una propiedad 'font-size' y 'line-height', incluso si estas no son utilizadas directamente para determinar la altura de la caja. Sin embargo, 'font-size' es, utilizada para definir las unidades 'em' y 'ex', y 'line-height' tiene un papel en la propiedad 'vertical-align'.

Como sigue:

Observe que los elementos reemplazados tienen una propiedad 'font-size' y 'line-height', incluso si estas no utilizan directamente la altura de la caja: los valores 'em' y 'ex' son relativos a los valores de 'font-size' y los valores en porcentajes para 'vertical-align' son relativos a los valores de 'line-height'.

Bajo 'line-height', después de la frase "Si la propiedad es fijada sobre un elemento a nivel de bloque cuyo contenido está compuesto por elementos a nivel de línea, se especifica la altura mínima de cada caja en línea generada," añadiendo la siguiente aclaración:

La altura mínima consiste en la altura mínima encima de la línea base del bloque y una profundidad mínima debajo de ella, exactamente como si cada línea comenzará con una anchura cero de la caja en línea con la fuente del bloque y las propiedades de la altura de línea (Este T_EX se llama "strut").

C.4.18 Sección 11.1

Aclaración de los dos últimos puntos cuando ocurre un desbordamiento:

- Una caja descendente es posicionada absolutamente parcialmente fuera de la caja.
- Una caja descendente tiene márgenes negativos, causando estos que sea posicionada parcialmente fuera de la caja.

C.4.19 Sección 11.1.1

Se quita 'proyección' para esta frase bajo el valor 'scroll'

Cuando este valor es especificado y el medio objetivo es la 'impresión' o 'proyección', el contenido desbordado deben ser impreso.

C.4.20 Sección 11.1.2

Los valores de "rect()" deben ser separados por comas. Así, la definición de <shape> ahora empieza:

En CSS2, el único valor válido de <shape> es: rect (<top>, <right>, <bottom>, <left>) ...

Debido a esta ambigüedad, las aplicaciones del usuario pueden soportar separaciones en "rect()" con o sin comas.

C.4.21 Sección 12.1 Los pseudo-elementos :before y :after

Aclaración de las líneas siguientes:

Los pseudo-elementos :before y :after permiten a los elementos los valores de la propiedad 'display' como sigue:

- Si el sujeto del selector es un elemento a nivel de bloque, están permitidos los valores 'none', 'inline' y 'block'. Si el valor de la propiedad 'display' de los pseudo-elementos tienen cualquier otro valor, el pseudo-elemento se comportará como si su valor fuese 'block'.
- Si el sujeto del selector es un elemento a nivel de línea, son permitidos los valores 'none' y 'inline'. Si el valor de la propiedad 'display' de los pseudo-elementos tiene cualquier otro valor, el pseudo-elemento se comparará como si su valor fuese 'inline'.

C.4.22 Sección 12.3.2 Inserción de comillas con la propiedad 'content'

Se añade la siguiente frase al final del segundo párrafo:

Una 'close-quote' que se hiciera de profundidad negativa sería un error y es ignorada (en tiempo de proceso): la profundidad permanece en 0 y no se procesa rinde ninguna comilla (aunque el resto del valor de la propiedad 'content' todavía se inserta).

C.4.23 Sección 12.5.1 Listas

Para aclarar la numeración Hebrea, se añade "(Alef, Bet, ... Tet Vav, Tet Zayin, ... Yod Tet, Kaf ...)".

C.4.24 Sección 14.2 El fondo

En la segunda oración: "En términos del modelo de caja, el "background" se refiere al fondo de las áreas del contenido y de relleno" ahora también menciona el área del borde. (Ver también [erratas de la Sección 8.1](#) arriba.) así:

En términos del modelo de caja, el "background" se refiere al fondo de las áreas del contenido, de relleno y de bordes.

En el cuarto párrafo, se añade al final de "Las aplicaciones del usuario deben observar las siguientes reglas de precedencia para rellenar el fondo" las siguientes palabras: "de las capas".

C.4.25 Sección 14.2.1 Propiedades del fondo

Se añade esta nota después del primer párrafo después de 'background-attachment':

Observe que hay solamente un acceso visual por documento. Es decir, incluso si un elemento tiene un mecanismo de desplazamiento (ver 'overflow'), un fondo 'fixed' no se mueve con el.

Bajo 'background-repeat', la frase "Todo el mosaico cubre el área de contenido y de relleno [...]" ha sido corregida a "Todo el mosaico cubre el área de contención, relleno **y borde** [...]".

Bajo 'background-attachment', la frase "incluso si la imagen es fija [...] el área del fondo o del relleno del elemento" ha sido corregida a

Incluso si la imagen es fija, sigue siendo visible cuando esta en el área del fondo, del relleno o del borde del elemento.

C.4.26 Sección 16.1

Añadido a:

El valor de 'text-indent' puede ser negativo, pero puede haber límites específicos de la implementación.

La siguiente aclaración: "Si el valor de 'text-indent' es negativo, el valor de 'overflow' afectará si el texto es visible."

C.4.27 Sección 16.2 Alineación: la propiedad 'text-align'

Se cambia "double justify" a "justify" bajo "left, right, center, y justify".

C.4.28 Sección 17.5.1 Capas y transparencias de tabla

En el punto 6, se cambia 'Estas celdas "vacías" son transparentes' a:

Si el valor de su propiedad 'empty-cells' es 'hide' estas celdas "vacías" son dejando ver el fondo de la celda, la fila, el grupo de filas, la columna y grupo de columnas, mostrando el fondo de la tabla

Para quitar la ambigüedad sobre la posición del fondo sobre filas y columnas, el siguiente párrafo fue añadido después del punto 6:

Los límites de las filas, columnas, grupos de filas y grupos de columnas en el [modelo de bordes cerrados](#) coincide con las hipotéticas líneas de la rejilla en la que los bordes de las celdas son centradas. (y así, en este modelo, las filas juntas cubren exactamente la tabla, no dejando ningún hueco, lo mismo con las columnas) en el [modelo de bordes separados](#), el límite coincide con el [límite del borde](#) de las celdas. (y así, en este modelo puede haber huecos entre las filas, las columnas, los grupos de filas o los grupos de columnas, correspondiendo a la propiedad '[border-spacing](#)')

Al final de la sección se añade el siguiente párrafo:

Observe que si la tabla tiene 'border-collapse: separate', el fondo del área dada por la propiedad '[border-spacing](#)' es siempre el fondo del elemento table. Ver el 17.6.1

C.4.29 Sección 17.5.2 Algoritmos para el ancho de la tabla

Se añade el siguiente párrafo después del párrafo inicial en esta sección:

Observe que esta sección anula las reglas que se aplican para calcular los anchos como se describe en la [sección 10.3](#). En particular, si los márgenes de la tabla son fijados a '0' y el ancho a 'auto', la tabla no pondrá el tamaño automáticamente para llenar su bloque de contención. Sin embargo, una vez el valor computado de 'width' para la tabla es encontrado (usando los algoritmos dados abajo o, cuando sea necesario, algún otro algoritmo dependiente de la AU) entonces otras partes de la sección 10.3 se aplican. Por otro lado la tabla puede ser centrada usando los márgenes izquierda y derecha como 'auto', por ejemplo.

El WG puede introducir los métodos para hacer automáticamente tablas encajando su bloque de contención en CSS3.

C.4.30 Sección 17.6.1 El modelo de bordes separados

Se añade una aclaración sobre la alineación del fondo de las filas/columnas. La frase "Este espacio se mantiene con el fondo del elemento tabla " fue sustituido por:

En este espacio, el fondo de la fila, columna, grupo de filas y grupo de columnas es invisible, permitiendo que el fondo de la tabla sea mostrado a través de estos.

C.4.31 Bordes alrededor de celdas vacías: la propiedad 'empty-cells'

La propiedad 'empty-cells' no solamente controla el borde, también el fondo.

C.4.32 Sección 17.6.2 El modelo de bordes cerrados

En la frase después de la pregunta, se añade " y padding-left_i y padding-right_i se refieren al relleno izquierdo (respectivamente, derecho) de la celda i."

C.4.33 Sección 18.2

Para el valor de 'ButtonHighlight', se cambio la descripción de "sombra oscura" a "color de luz más clara".

C.4.34 Sección A.3

La frase entre paréntesis "algo análogo a la propiedad 'display'" fue engañoso. La propiedad 'speak' se parece a 'visibility' en algunas cosas y a 'display' en otras.

C.4.35 Apéndice G.2 Escaner léxico

Se quita la siguiente línea del escáner como sino apareciera en la gramática:

```
"@" {ident}           {return ATKEYWORD;}
```

El comando está en el escáner para asegurar que un numero seguido por un identificador es leído como un comando en lugar de dos. Este caso es considerado un error en CSS2.

C.4.36 Apéndice E. Referencias

La entrada para "[URI]" referido a un esbozo que se ha hecho en RFC. La entrada ha sido cambiada.

Apéndice D Hoja de estilo por defecto para HTML 4.0

Este apéndice es informativo, no normativo.

Esta hoja de estilo describe el formato típico de todos los elementos de HTML 4.0 ([\[HTML40\]](#)) basados en la investigación extensa que las AU actuales practican. Se anima a los desarrolladores que lo utilicen como una hoja de estilo por defecto en sus implementaciones.

La presentación completa de algunos elementos de HTML no se puede expresar en CSS 2.1, incluyendo [los elementos](#) sustituidos ("img", "object"), los elementos scripting ("script", "applet"), los elementos del control de la forma, y los elementos del marco.

Para otros elementos, la presentación de la herencia se puede describir en CSS pero la solución quita el elemento. Por ejemplo, el elemento FONT se puede sustituir por declaraciones de atadura de CSS a otros elementos (Ej., DIV). Asimismo, la presentación de la herencia atributos de presentación (Ej., el atributo "borde" en la TABLE) se puede describir en CSS, pero el margen de beneficio en el documento fuente debe ser cambiado.

```
html, address,
blockquote,
body, dd, div,
dl, dt, fieldset, form,
frame, frameset,
h1, h2, h3, h4,
h5, h6, noframes,
ol, p, ul, center,
dir, hr, menu, pre    { display: block }
li                    { display: list-item }
head                  { display: none }
table                 { display: table }
tr                    { display: table-row }
thead                 { display: table-header-group }
tbody                 { display: table-row-group }
tfoot                 { display: table-footer-group }
col                   { display: table-column }
colgroup              { display: table-column-group }
td, th                { display: table-cell; }
caption               { display: table-caption }
th                    { font-weight: bold; text-align: center }
caption               { text-align: center }
body                  { margin: 8px; line-height: 1.12 }
h1                    { font-size: 2em; margin: .67em 0 }
h2                    { font-size: 1.5em; margin: .75em 0 }
h3                    { font-size: 1.17em; margin: .83em 0 }
h4, p,
blockquote, ul,
fieldset, form,
ol, dl, dir,
menu                  { margin: 1.12em 0 }
h5                    { font-size: .83em; margin: 1.5em 0 }
h6                    { font-size: .75em; margin: 1.67em 0 }
h1, h2, h3, h4,
h5, h6, b,
```

```

strong          { font-weight: bolder }
blockquote      { margin-left: 40px; margin-right: 40px }
i, cite, em,
var, address    { font-style: italic }
pre, tt, code,
kbd, samp       { font-family: monospace }
pre             { white-space: pre }
button, textarea,
input, object,
select          { display:inline-block; }
big             { font-size: 1.17em }
small, sub, sup { font-size: .83em }
sub             { vertical-align: sub }
sup             { vertical-align: super }
table           { border-spacing: 2px; }
thead, tbody,
tfoot          { vertical-align: middle }
td, th         { vertical-align: inherit }
s, strike, del { text-decoration: line-through }
hr             { border: 1px inset }
ol, ul, dir,
menu, dd       { margin-left: 40px }
ol             { list-style-type: decimal }
ol ul, ul ol,
ul ul, ol ol   { margin-top: 0; margin-bottom: 0 }
u, ins         { text-decoration: underline }
br:before      { content: "\A" }
:before, :after { white-space: pre-line }
center         { text-align: center }
abbr, acronym  { font-variant: small-caps; letter-spacing: 0.1em }
:link, :visited { text-decoration: underline }
:focus         { outline: thin dotted invert }

/* Begin bidirectionality settings (do not change) */
BDO[DIR="ltr"] { direction: ltr; unicode-bidi: bidi-override }
BDO[DIR="rtl"] { direction: rtl; unicode-bidi: bidi-override }

*[DIR="ltr"]   { direction: ltr; unicode-bidi: embed }
*[DIR="rtl"]   { direction: rtl; unicode-bidi: embed }

@media print {
  h1           { page-break-before: always }
  h1, h2, h3,
  h4, h5, h6   { page-break-after: avoid }
  ul, ol, dl   { page-break-before: avoid }
}

```

Apéndice E Descripción de elaboración de contextos apilados

Contenido

- [E.1 Definiciones](#)
- [E.2 Orden de la pintura](#)
- [E.3 Notas](#)

Este capítulo define la orden de la pintura de CSS2.1 más detalladamente que la descrita en el resto de la especificación.

E.1 Definiciones

Orden de la estructura

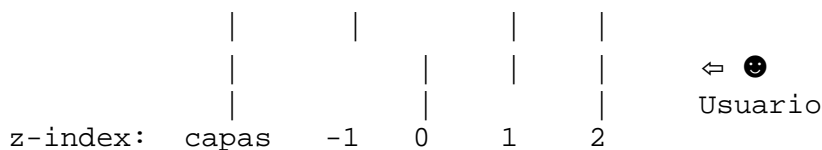
Preorden de la primera profundidad transversal del *procesamiento de la estructura* (árbol), en orden lógico (no visual) para el contenido bidireccional, después de considerar las propiedades que mueven las cajas alrededor tales como el valor 'run-in' de 'display'.

Elemento

En esta descripción, el "elemento" se refiere a ambos elementos reales y pseudo-elementos. Los Pseudo-elementos y las cajas anónimas se tratan como descendientes en los lugares apropiados. Por ejemplo, un marcador externo de la lista viene antes que una caja adyacente ':before' en la caja de línea, que viene antes del contenido de la caja, y así sucesivamente.

E.2 Orden de la pintura

La parte inferior del apilado está mas alejado del usuario, la parte superior del apilado está mas cerca del usuario. El fondo del apilado es el más futuro del usuario, la tapa del apilado es el más cercano al usuario:



El fondo del contexto apilado y el contexto apilado posicionado mas negativo son la pila inferior (del fondo), mientras que el contexto apilado posicionado mas positivo son la pila superior (de arriba).

Las capas son transparentes si están contenidas dentro de otra, y la AU le da un color definido si no lo es. Es infinita en grado y contiene el elemento de la raíz. Inicialmente, el acceso visual se ancla con su esquina superior izquierda en el origen de las capas.

El orden de apilamiento para un elemento generando un contexto de apilamiento (Vea la propiedad 'z-index') es:

1. Si el elemento es un elemento raíz:
 1. El color del fondo del elemento sobre las capas entera.
 2. La imagen de fondo del elemento, sin acortada, pintada en el origen que sería utilizada si fuese pintada para el elemento raíz.
2. Si el elemento es un bloque, un elemento lista, u otro bloque equivalente:
 1. El color del fondo del elemento a menos que sea el elemento de la raíz.
 2. La imagen de fondo del elemento a menos que sea el elemento de la raíz.
 3. El borde del elemento.

Si no, si el elemento es una tabla a nivel de bloque:

4. Los fondos de la tabla (color o imagen) a menos que sea el elemento raíz.
 5. Los fondos del grupo de columna (color o imagen).
 6. Los fondos de la columna (color o imagen).
 7. Los fondos del grupo de fila (color o imagen).
 8. Los fondos de la fila (color o imagen).
 9. Los fondos de la celda (color o imagen).
 10. Todos los bordes de la tabla (en el orden de la estructura para bordes separados).
3. Apilar contextos formados por los descendientes colocados con z-index negativos (excepto 0) en orden del z-index (primero el más negativo) luego en orden de la estructura.
 4. Para toda sus descendentes en flujo, no posicionados, a nivel de bloque el orden de la estructura: Si el elemento es un bloque, un elemento lista, u otro bloque equivalente:
 1. El color del fondo del elemento.
 2. La imagen de fondo del elemento.
 3. El borde del elemento.

Si no, si el elemento es una tabla:

4. Los fondos de la tabla (color o imagen).
 5. Los fondos del grupo de columna (color o imagen).
 6. Los fondos de la columna (color o imagen).
 7. Los fondos del grupo de fila (color o imagen).
 8. Los fondos de la fila (color o imagen).
 9. Los fondos de la celda (color o imagen).
 10. Todos los bordes de la tabla (en el orden para la estructura de bordes separados).
5. Todos los descendientes flotantes, en orden a la estructura (árbol). Para cada uno de éstos, trate el elemento como si creara un nuevo contexto apilado, pero cualquier descendiente que cree realmente un nuevo contexto de pila debe ser considerado parte del contexto que apila el padre, no uno nuevo.
 6. Primero para el elemento, entonces para todos sus descendentes en flujo, no posicionados, a nivel de bloque el orden la estructura:
 1. Para cada elemento en cada caja de línea del elemento, si lo hay, o si el elemento es a nivel de línea (observe que todos los elementos en cajas de línea son forzosamente a nivel de línea), sobre una base por línea de caja:

1. Color del fondo del elemento.
2. Imagen de fondo del elemento.
3. Borde del elemento.
4. Para los elementos en línea:
 1. Subrayado del elemento.
 2. Sobre rayado del elemento.
 3. Texto del elemento.
 4. Luego los descendentes en flujo, no posicionados, a nivel de línea de elementos, incluyendo los elementos anónimos en línea generados para los nodos del texto dentro del elemento, en orden de estructura (árbol):
 1. Salte al paso 6.1 para este elemento.
 5. A través de la línea del elemento.

Para bloques en línea (inline-block) y los elementos de tabla en línea (inline-table):

6. Para cada uno de éstos, trate al elemento como si creara un nuevo contexto de pila, pero cualquier descendiente que cree realmente un nuevo contexto de pila debe ser considerado parte del contexto que apila el padre, no uno nuevo.

Para elementos a nivel de línea reemplazados:

7. El contenido reemplazado, atómico.
5. Opcionalmente, el contorno del elemento (véase el 9 abajo).
2. Si no, si el elemento es un elemento a nivel de bloque reemplazado, después el contenido reemplazado, atómico.
3. Opcionalmente, si el elemento es a nivel de bloque, el contorno del elemento (véase 9 abajo).
7. Todos los descendientes posicionados 'z-index: auto' o 'z-index: 0', en orden de la estructura. Para éstos 'z-index: auto', trate al elemento como si creara un nuevo contexto de pila, pero cualquier descendiente que cree realmente un nuevo contexto de pila debe ser considerado parte del contexto que apila el padre, no uno nuevo. Para éstos 'z-index: 0', trate el contexto de pila generado atómico.
8. Apilar contextos formados por los descendientes posicionados con los z-index mayores que o igual a 1 en orden del z-index (primero la de valor mas bajo) de la estructura
9. Finalmente, las implementaciones que no dibujan contornos en los pasos anteriores deben dibujar contornos de este contexto de pila en esta etapa.

E.3 Notas

El fondo del elemento raíz se pinta solamente una vez, sobre el lienzo entero.

Mientras que los fondos de en líneas bidireccionales se pintan en orden a la estructura, se posicionan en orden visual. Puesto que la colocación de fondos en línea está sin especificar en CSS2.1, el resultado exacto de estos dos requisitos lo define la AU. CSS3 puede definir esto más detalladamente.

Apéndice F Tabla completa de las propiedades

Este apéndice es informativo, no normativo.

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
' azimuth '	<ángulo> [[left-side far-left left center-left center center-right right far-right right-side] behind] leftwards rightwards inherit	center		si		auditivos
' background-attachment '	scroll fixed inherit	scroll		no		visual
' background-color '	<color> transparent inherit	transparent		no		visual
' background-image '	<uri> none inherit	none		no		visual
' background-position '	[[<porcentaje> <medida> left center right] [<porcentaje> <medida> top center bottom]?] [[left center right] [top center bottom]] inherit	0% 0%		no	referido al tamaño de la caja en si	visual
' background-repeat '	repeat repeat-x repeat-y no-repeat inherit	repeat		no		visual
' background '	[' background-color ' ' background-image ' ' background-repeat ' ' background-attachment ' ' background-position '] inherit	ver las propiedades individuales		no	permitido sobre ' background-position '	visual
' border-collapse '	collapse separate inherit	separate	elementos 'table' e 'inline-table'	si		visual
' border-color '	[<color> transparent]{1,4} inherit	ver las propiedades individuales		no		visual
' border-spacing '	<medida> <medida>? inherit	0	elementos 'table' e 'inline-table'	si		visual

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'border-style'	<border-style>{1,4} inherit	ver las propiedades individuales		no		visual
'border-top' 'border-right' 'border-bottom' 'border-left'	[<border-width> <border-style> 'border-top-color'] inherit	ver las propiedades individuales		no		visual
'border-top-color' 'border-right-color' 'border-bottom-color' 'border-left-color'	<color> transparent inherit	El valor de la propiedad 'color'		no		visual
'border-top-style' 'border-right-style' 'border-bottom-style' 'border-left-style'	<border-style> inherit	none		no		visual
'border-top-width' 'border-right-width' 'border-bottom-width' 'border-left-width'	<border-width> inherit	medium		no		visual
'border-width'	<border-width>{1,4} inherit	ver las propiedades individuales		no		visual
'border'	[<border-width> <border-style> 'border-top-color'] inherit	ver las propiedades individuales		no		visual
'bottom'	<medida> <porcentaje> auto inherit	auto	Elementos posicionados	no	Referido a la altura del bloque de contención	visual
'caption-side'	top bottom inherit	top	Elementos 'table-caption'	si		visual
'clear'	none left right both inherit	none	Elementos a nivel de bloque	no		visual
'clip'	<shape> auto inherit	auto	Elementos posicionados absolutamente	no		visual

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'color'	<color> inherit	Depende de la aplicación del usuario		si		visual
'content'	normal [<string> <uri> <contador> attr(<identificador>) open-quote close-quote no-open-quote no-close-quote]+ inherit	normal	Pseudo-elementos :before y :after	no		todos
'contador-increment'	[<identificador> <entero>?]+ none inherit	none		no		todos
'contador-reset'	[<identificador> <entero>?]+ none inherit	none		no		todos
'cue-after'	<uri> none inherit	none		no		auditivos
'cue-before'	<uri> none inherit	none		no		auditivos
'cue'	['cue-before' 'cue-after'] inherit	ver propiedades individuales		no		auditivos
'cursor'	[[<uri> ,]* [auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress]] inherit	auto		si		visual, interactivo
'direction'	ltr rtl inherit	ltr	todos los elementos, pero vea el texto	si		visual
'display'	inline block list-item run-in inline-block table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption none inherit	inline		no		todos

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'elevation'	<ángulo> below level above higher lower inherit	level		si		auditivos
'empty-cells'	show hide inherit	show	Elementos 'table-cell'	si		visual
'float'	left right none inherit	none	todos, pero vea 9.7	no		visual
'font-family'	[[<nombre-familia> <familia-generica>] [,<nombre-familia> <familia-generica>]*] inherit	depende de la aplicación del usuario		si		visual
'font-size'	<tamaño-absol> <tamaño-relativo> <medida> <percentage> inherit	medium		si	Referido al tamaño de la fuente del elemento padre	visual
'font-style'	normal italic oblique inherit	normal		si		visual
'font-variant'	normal small-caps inherit	normal		si		visual
'font-weight'	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	normal		si		visual
'font'	[['font-style' 'font-variant' 'font-weight']? 'font-size' [/ 'line-height']? 'font-family'] caption icon menu message-box small-caption status-bar inherit	ver las propiedades individuales		si	ver las propiedades individuales	visual
'height'	<medida> <porcentaje> auto inherit	auto	todos elementos pero elementos no reemplazados en línea, columnas de tablas, y grupos de columnas	no	vea el texto	visual
'left'	<medida> <porcentaje> auto inherit	auto	elementos posicionados	no	Referido al ancho del bloque de contención	visual
'letter-spacing'	normal <medida> inherit	normal		si		visual

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'line-height'	normal <número> <medida> <porcentaje> inherit	normal		si	Referido al tamaño de la fuente del propio elemento	visual
'list-style-image'	<uri> none inherit	none	elementos con 'display: list-item'	si		visual
'list-style-position'	inside outside inherit	outside	elementos con 'display: list-item'	si		visual
'list-style-type'	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian none inherit	disc	elementos con 'display: list-item'	si		visual
'list-style'	['list-style-type' 'list-style-position' 'list-style-image'] inherit	ver las propiedades individuales	elementos con 'display: list-item'	si		visual
'margin-right' 'margin-left'	<margin-width> inherit	0	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'	no	Referido al ancho del bloque de contención	visual
'margin-top' 'margin-bottom'	<margin-width> inherit	0	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'	no	Referido al ancho del bloque de contención	visual
'margin'	<margin-width>{1,4} inherit	ver las propiedades individuales	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table'	no	Referido al ancho del bloque de contención	visual
'max-height'	<medida> <porcentaje> none inherit	none	todos los elementos excepto elementos no reemplazados en línea y elementos tabla	no	vea el texto	visual

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
' max-width '	<medida> <porcentaje> none inherit	none	todos los elementos excepto elementos no reemplazados en línea y elementos tabla	no	Referido al ancho del bloque de contención	visual
' min-height '	<medida> <porcentaje> inherit	0	todos los elementos excepto elementos no reemplazados en línea y elementos tabla	no	vea el texto	visual
' min-width '	<medida> <porcentaje> inherit	0	todos los elementos excepto elementos no reemplazados en línea y elementos tabla	no	Referido al ancho del bloque de contención	visual
' orphans '	<entero> inherit	2	Elementos a nivel de bloque	si		visual, paginados
' outline-color '	<color> invert inherit	invert		no		visual, interactivo
' outline-style '	<border-style> inherit	none		no		visual, interactivo
' outline-width '	<border-width> inherit	medium		no		visual, interactivo
' outline '	[' outline-color ' ' outline-style ' ' outline-width '] inherit	ver las propiedades individuales		no		visual, interactivo
' overflow '	visible hidden scroll auto inherit	visible	Elementos a nivel de bloque y reemplazados, celdas de tabla, bloques en línea	no		visual
'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	<padding-width> inherit	0	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table' y 'table-cell'	no	refer to width of containing block	visual

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'padding'	<padding-width>{1,4} inherit	ver las propiedades individuales	todos los elementos excepto elementos tabla con display distinto a 'table' o 'inline-table' y 'table-cell'	no	Referido al ancho del bloque de contención	visual
'page-break-after'	auto always avoid left right inherit	auto	elementos a nivel de bloque	no		visual, paginados
'page-break-before'	auto always avoid left right inherit	auto	elementos a nivel de bloque	no		visual, paginados
'page-break-inside'	avoid auto inherit	auto	elementos a nivel de bloque	si		visual, paginados
'pause-after'	<tiempo> <porcentaje> inherit	0		no	vea el texto	auditivos
'pause-before'	<tiempo> <porcentaje> inherit	0		no	vea el texto	auditivos
'pause'	[[<tiempo> <porcentaje>]{1,2}] inherit	ver las propiedades individuales		no	Ver la descripción de 'pause-before' y 'pause-after'	auditivos
'pitch-range'	<número> inherit	50		si		auditivos
'pitch'	<frecuencia> x-low low medium high x-high inherit	medium		si		auditivos
'play-during'	<uri> [mix repeat]? auto none inherit	auto		no		auditivos
'position'	static relative absolute fixed inherit	static		no		visual
'quotes'	[<cadena> <cadena>]+ none inherit	depende de la aplicación del usuario		si		visual
'richness'	<número> inherit	50		si		auditivos
'right'	<medida> <porcentaje> auto inherit	auto	elementos posicionados	no	Referido al ancho del bloque de contención	visual
'speak-header'	once always inherit	once	Elementos que tienen información en la cabecera de la tabla	si		auditivos
'speak-numeral'	dígito continuo inherit	continuo		si		auditivos

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
' speak-punctuation '	code none inherit	none		si		auditivos
' speak '	normal none spell-out inherit	normal		si		auditivos
' speech-rate '	<número> x-slow slow medium fast x-fast faster slower inherit	medium		si		auditivos
' stress '	<número> inherit	50		si		auditivos
' table-layout '	auto fixed inherit	auto	elementos 'table' e 'inline-table'	no		visual
' text-align '	left right center justify inherit	'left' si 'direction' es 'ltr'; 'right' si 'direction' es 'rtl'	elementos a nivel de bloque, table cells and inline blocks	si		visual
' text-decoration '	none [underline overline line-through blink] inherit	none		no (vea el texto)		visual
' text-indent '	<medida> <porcentaje> inherit	0	elementos a nivel de bloque, table cells and inline blocks	si	Referido al ancho del bloque de contención	visual
' text-transform '	capitalize uppercase lowercase none inherit	none		si		visual
' top '	<medida> <porcentaje> auto inherit	auto	elementos posicionados	no	Referidos a la altura del bloque de contención	visual
' unicode-bidi '	normal embed bidi-override inherit	normal	todos los elementos, pero vea el texto	no		visual
' vertical-align '	baseline sub super top text-top middle bottom text-bottom <porcentaje> <medida> inherit	baseline	Elementos a nivel de línea y 'table-cell'	no	Referido a la 'altura de la línea' del elemento en sí	visual
' visibility '	visible hidden collapse inherit	visible		si		visual
' voice-family '	[[<voz-específica> <voz-genérica>],]* [<voz-específica> <voz-genérica>] inherit	depende de la aplicación del usuario		si		auditivos

Nombre	Valores	Valor Inicial	Se aplica a (Predeterminado: todos)	Se hereda ?	Porcentajes (Predeterminado: N/A)	Grupo de medios
'volume'	<número> <porcentaje> silent x-soft soft medium loud x-loud inherit	medium		si	Referido al valor heredado	auditivos
'white-space'	normal pre nowrap pre-wrap pre-line inherit	normal		si		visual
'widows'	<entero> inherit	2	elementos a nivel de bloque	si		visual, paginados
'width'	<medida> <porcentaje> auto inherit	auto	todos elements but non-replaced inline elements, table rows, and row groups	no	Referidos al ancho del bloque de contención	visual
'word-spacing'	normal <medida> inherit	normal		si		visual
'z-index'	auto <entero> inherit	auto	elementos posicionados	no		visual

Apéndice G. Gramática de CSS 2.1

Contenido

- [G.1 Gramática](#)
- [G.2 Escaner léxico](#)
- [G.3 Comparación de comandos en CSS 2.1 y CSS1](#)

Este apéndice es normativo.

La gramática de abajo define la sintaxis de CSS 2.1. sin embargo, es en cierto sentido, un sobreconjunto de CSS 2.1 mientras que esta especificación impone las restricciones semánticos adicionales no expresados en esta gramática. Una AU conformada debe también adherir a las [reglas de análisis con compatibilidad futura](#), a la notación de los selectores, a la [notación de propiedades y valores](#), y a la unidad de notación. Además, el lenguaje del documento puede imponer restricciones, ej., HTML impone restricciones ante los valores posibles del atributo "class".

G.1 Gramática

La gramática de abajo es LALR(1) (pero observe que la mayoría de las AU no deben utilizarla directamente, puesto que no expresa las [convenciones de análisis](#), solamente la sintaxis de CSS2.1). El formato de las producciones se optimiza para el uso humano y se utiliza alguna notación abreviada más allá de Yacc (véase [YACC](#)):

- * : 0 o más
- + : 1 o más
- ? : 0 o 1
- | : separa alternativas
- [] : agrupar

Las producciones son:

```
stylesheet
  : [ CHARSET_SYM S* STRING S* ';' ]?
    [S|CDO|CDC]* [ import [S|CDO|CDC]* ]*
    [ [ ruleset | media | page ] [S|CDO|CDC]* ]*
  ;
import
  : IMPORT_SYM S*
    [STRING|URI] S* [ medium [ COMMA S* medium]* ]? ';' S*
  ;
media
  : MEDIA_SYM S* medium [ COMMA S* medium ]* LBRACE S* ruleset* '}' S*
  ;
medium
  : IDENT S*
  ;
page
  : PAGE_SYM S* pseudo_page? S*
    LBRACE S* declaration [ ';' S* declaration ]* '}' S*
```

```

;
pseudo_page
: ':' IDENT
;
operator
: '/' S* | COMMA S* | /* empty */
;
combinator
: PLUS S*
| GREATER S*
| S+
;
unary_operator
: '-' | PLUS
;
property
: IDENT S*
;
ruleset
: selector [ COMMA S* selector ]*
  LBRACE S* declaration [ ';' S* declaration ]* '}' S*
;
selector
: simple_selector [ combinator simple_selector ]*
;
simple_selector
: element_name [ HASH | class | attrib | pseudo ]*
| [ HASH | class | attrib | pseudo ]+
;
class
: '.' IDENT
;
element_name
: IDENT | '*'
;
attrib
: '[' S* IDENT S* [ [ '=' | INCLUDES | DASHMATCH ] S*
  [ IDENT | STRING ] S* ]? ']'
;
pseudo
: ':' [ IDENT | FUNCTION S* IDENT? S* ')' ]
;
declaration
: property ':' S* expr prio?
| /* empty */
;
prio
: IMPORTANT_SYM S*
;
expr
: term [ operator term ]*
;
term
: unary_operator?
  [ NUMBER S* | PERCENTAGE S* | LENGTH S* | EMS S* | EXS S* | ANGLE S* |
  TIME S* | FREQ S* | function ]

```

```

    | STRING S* | IDENT S* | URI S* | hexcolor
;
function
  : FUNCTION S* expr ')' S*
;
/*
 * There is a constraint on the color that it must
 * have either 3 or 6 hex-digits (i.e., [0-9a-fA-F])
 * after the "#"; e.g., "#000" is OK, but "#abcd" is not.
 */
hexcolor
  : HASH S*
;

```

G.2 Escaner léxico

Lo siguiente son los comandos (tokenizer), escrito en notación Flex (véase [FLEX](#)). Los comandos son sensibles a mayúsculas y minúsculas.

Las dos apariciones "\377" representan el número más alto del carácter que la versión actual de Flex puede manejar (decimal 255). Deben ser leídas como "\4177777" (decimal 1114111), que es el punto de código posible más alto en Unicode/Iso-10646 .

```

%option case-insensitive

h          [0-9a-f]
nonascii  [\200-\377]
unicode    \\{h}{1,6}(\r\n|[ \t\r\n\f])?
escape     {unicode}|\|[ \t\r\n\f]
nmstart    [_a-z]|{nonascii}|{escape}
nmchar     [_a-zA-Z0-9-]|{nonascii}|{escape}
string1    \"([ \t !#$%&(-~)|\\{nl}|\\'|{nonascii}|{escape})*\"
string2    \'([ \t !#$%&(-~)|\\{nl}|\\\"|{nonascii}|{escape})*\'

ident      {nmstart}{nmchar}*
name       {nmchar}+
num        [0-9]+|[0-9]*\".[0-9]+
string     {string1}|{string2}
url        (!#$%&*~)|{nonascii}|{escape})*
s          [ \t\r\n\f]
w          {s}*
nl         \n|\r\n|\r|\f
range      \{1,6\}|{h}(\{0,5\}|{h}(\{0,4\}|{h}(\{0,3\}|{h}(\{0,2\}|{h}(\{0,1\}|{h}))))
))

%%

{s}+      {return S;}

\\/*[^\*]*\*+([^\*]*[^\*]*\*+)*\\/ /* ignore comments */

"<!--"    {return CDO;}
"-->"     {return CDC;}

```

```

"~="          {return INCLUDES;}
"|="          {return DASHMATCH;}

{w}"{"       {return LBRACE;}
{w}"+"       {return PLUS;}
{w}">"       {return GREATER;}
{w}","       {return COMMA;}

{string}      {return STRING;}

{ident}       {return IDENT;}

"#"{name}    {return HASH;}

"@import"     {return IMPORT_SYM;}
"@page"       {return PAGE_SYM;}
"@media"      {return MEDIA_SYM;}
"@charset"    {return CHARSET_SYM;}

"!"{w}"important" {return IMPORTANT_SYM;}

{num}em       {return EMS;}
{num}ex       {return EXS;}
{num}px       {return LENGTH;}
{num}cm       {return LENGTH;}
{num}mm       {return LENGTH;}
{num}in       {return LENGTH;}
{num}pt       {return LENGTH;}
{num}pc       {return LENGTH;}
{num}deg      {return ANGLE;}
{num}rad      {return ANGLE;}
{num}grad     {return ANGLE;}
{num}ms       {return TIME;}
{num}s        {return TIME;}
{num}Hz       {return FREQ;}
{num}kHz      {return FREQ;}
{num}{ident}  {return DIMEN;}
{num}%        {return PERCENTAGE;}
{num}         {return NUMBER;}

"url("{w}{string}{w}")" {return URI;}
"url("{w}{url}{w}")"    {return URI;}
{ident}"("             {return FUNCTION;}

.                       {return *yytext;}

```

G.3 Comparación de los comandos en CSS2.1 y CSS1

Hay algunas diferencias en la sintaxis especificada en la recomendación CSS1 ([\[CSS1\]](#)), y la de arriba. La mayoría de éstas son debido a los nuevos comandos en CSS2 que no existían en CSS1. Otras son debido a que la gramática se ha reescrito para ser más legible. Sin embargo, hay algunos cambios incompatibles, que eran considerados errores en la sintaxis de CSS1. Se explican abajo.

- Las hojas del estilo CSS1 solo podían estar en una codificación de 1 byte por carácter, tales como ASCII e Iso-8859-1. CSS2.1 no tiene tal limitación. En la práctica, había pequeñas dificultades en la extrapolación de los comandos de CSS1, y algunas AUs han aceptado la codificación de 2 byte.
- CSS1 permitió solamente cuatro dígitos hexadecimales después de la barra invertida (\) para referirse a los caracteres de Unicode, CSS2 [permite seis](#). Además, CSS2 permite que un carácter de espacio en blanco delimite la sentencia de escape. Ej., según CSS1, la cadena "\abcdef" tiene 3 letras (\abcd, e, y f), según CSS2 tiene solamente una (\abcdef).
- El carácter del tabulador (ASCII 9) no estaba permitido en las cadenas. Sin embargo, puesto que las cadenas en CSS1 solo eran utilizadas para los nombres de las fuentes y para URLs, la única manera de que esto puede conducir a la incompatibilidad entre CSS1 y CSS2 es si una hoja de estilo contiene una familia de fuentes que tenga un tabulador en su nombre.
- De modo similar, los saltos de líneas ([escapados con una barra invertida](#)) no estaban permitidos en las cadenas en CSS1.
- CSS2 analiza un número seguido inmediatamente por un identificador como comando DIMEN (es decir, una unidad desconocida), CSS1 lo analizó como un número un identificador. Eso significa que en CSS1, la declaración 'font: 10pt/1.2serif' era correcta, al igual que 'font: serif 10pt/12pt'; en CSS2, se requiere un espacio antes "serif". (algunas AUs han aceptado el primer ejemplo, pero no el segundo.)
- En CSS1, un nombre de clase podría comenzar con un dígito ("55ft"), a menos que fuera una dimensión (".55in"). En CSS2, tales clases se analizan como dimensiones desconocidas (para permitir futuras adiciones de nuevas unidades). Para convertir "55ft" una clase válida, CSS2 requiere que el primer dígito sea escapado (".\35 5ft")

Apéndice I. Índice

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

- :active, [2](#)
- :after, [1](#), [2](#)
- :before, [1](#), [2](#)
- :first, [1](#)
- :first-child, [1](#)
- :first-letter, [1](#)
- :first-line, [1](#)
- :focus, [1](#)
- :hover, [1](#)
- :lang, [1](#)
- :left, [1](#)
- :link, [1](#)
- :right, [1](#)
- :visited, [1](#)
- =, [1](#)
- ~, [1](#)
- |, [1](#)

- @import, [1](#), [2](#), [3](#)
- @media, [1](#), [2](#)
- @page, [1](#)

- acceso visual, [1](#)
- active (pseudo-clase), [1](#)
- after, [1](#)
- análisis con compatibilidad futura, [1](#)
- <ángulo>, [1](#), [2](#)
 - definición de, [1](#)
- anónima, [1](#)
- antepasado, [1](#)
- aplicación del usuario, [1](#)
- área de alcance, [1](#)
- área de la página, [1](#)
- armenian, [1](#)
- asignando a los elementos partes de la tabla, [1](#)
- atributo, [1](#)
- attr(), [1](#)
- AU, [1](#)
- 'auditivos' grupo de medios, [1](#)
- 'azimuth', [1](#)

- 'background', [1](#)
- 'background-attachment', [1](#)
- 'background-color', [1](#)
- 'background-image', [1](#)
- 'background-position', [1](#)
- 'background-repeat', [1](#)
- before, [1](#)
- bidireccionalidad (bidi), [1](#)
- 'bitmap' grupo de medios, [1](#)
- 'block', definición de, [1](#)
- bloque, [1](#), [2](#)
- bloque de contención, [1](#), [2](#), [3](#)
 - inicial, [1](#)
- bloque de contención inicial, [1](#)
- bloque de declaraciones, [1](#)
- bloque en línea, [1](#), [2](#), [3](#), [4](#), [5](#)
- border
 - de una caja, [1](#)
- 'border', [1](#)
- 'border-bottom', [1](#)
- 'border-bottom-color', [1](#)
- 'border-bottom-style', [1](#)
- 'border-bottom-width', [1](#)
- 'border-collapse', [1](#)
- 'border-color', [1](#)
- 'border-left', [1](#)
- 'border-left-color', [1](#)
- 'border-left-style', [1](#)
- 'border-left-width', [1](#)
- 'border-right', [1](#)
- 'border-right-color', [1](#)
- 'border-right-style', [1](#)
- 'border-right-width', [1](#)
- 'border-spacing', [1](#)
- <border-style>, [1](#)
- <border-style>, definición de, [1](#)
- 'border-style', [1](#)
- 'border-top', [1](#)
- 'border-top-color', [1](#)
- 'border-top-style', [1](#)
- 'border-top-width', [1](#)
- <border-width>
 - definición de, [1](#)
- 'border-width', [1](#)
- bordes separados, [1](#)
- 'bottom', [1](#)
- <bottom>,
 - Definición de, [1](#)

- cadena, [1](#)
- <cadena>, [1](#), [2](#), [3](#)
- <cadena>, definición de, [1](#)
- caja
 - borde, [1](#)
 - contenido, [1](#)
 - altura del contenido, [1](#)
 - ancho del contenido, [1](#)
 - altura, [1](#)
 - margen, [1](#)
 - desborde, [1](#)
 - relleno, [1](#)
- caja de bloque, [1](#)
- caja de bloque principal, [1](#)
- caja de la página, [1](#)
- caja en línea, [1](#), [2](#)
- caja de línea, [1](#)
- cajas de línea anónimas, [1](#)
- capitulares, [1](#)
- capitulares caídas, [1](#)
- 'caption-side', [1](#)
- cascada, [1](#)
- circle, [1](#)
- 'clear', [1](#)
- 'clip', [1](#)
- close-quote, [1](#), [2](#)
- codificación de caracteres, [1](#)
 - predeterminada, [1](#)
 - aplicación del usuario determinación de, [1](#)
- código del lenguaje, [1](#)
- color, [1](#)
- <color>, [1](#), [2](#)
 - definición de, [1](#)
- 'color', [1](#)
- combinador, [1](#)
- comentarios, [1](#)
- conformidad, [1](#), [2](#), [3](#)
- <contador>, [1](#)
- <contador>, definición de, [1](#)
- contadores, [1](#)
- contenido, [1](#)
- contenido
 - de una caja, [1](#)
 - procesado, [1](#)
- contenido generado, [1](#)
- contenido procesado, [1](#)
- 'content', [1](#)
- contexto de página, [1](#)

- contexto de pila, [1](#)
- contexto de pila raíz, [1](#)
- contexto del formato, [1](#)
- 'continuos' grupo de medios, [1](#)
- contorno, [1](#)
- counter(), [1](#)
- 'counter-increment', [1](#)
- 'counter-reset', [1](#)
- 'cue', [1](#)
- 'cue-after', [1](#)
- 'cue-before', [1](#)
- 'cursor', [1](#)

- 'dashed', [1](#), [2](#)
- DEBE, [1](#)
- DEBERÁ, [1](#)
- DEBERÍA, [1](#)
- decimal, [1](#)
- decimal-leading-zero, [1](#)
- declaración, [1](#)
- declaraciones múltiples, [1](#)
- desborde, [1](#)
- descendiente, [1](#)
- dimensiones intrínsecas, [1](#)
- 'direction', [1](#)
- disc, [1](#)
- 'display', [1](#)
- documento fuente, [1](#)
- 'dotted', [1](#), [2](#)
- 'double', [1](#), [2](#)
- DTD, [1](#), [2](#)

- elemento, [1](#)
 - siguiente, [1](#)
 - precedente, [1](#)
- elemento a nivel de bloque, [1](#)
- elemento a nivel de línea, [1](#)
- elemento absolutamente posicionado, [1](#)
- elemento de la tabla, [1](#)
 - interno, [1](#)
- elemento de lista, [1](#)
- elemento interno de la tabla, [1](#)
- elemento precedente, [1](#)
- elemento reemplazado, [1](#)
- elemento siguiente, [1](#)
- elemento/caja posicionado/a, [1](#)

- 'elevation', [1](#)
- em (unidad), [1](#)
- 'empty-cells', [1](#)
- <entero>, [1](#)
 - definición de, [1](#)
- entidad del mensaje, [1](#)
- equivalencia, [1](#)
- equivalencia exacta, [1](#)
- equivalencia separada por espacios, [1](#)
- equivalencia separada por guiones, [1](#)
- escapes con barra invertida, [1](#)
- esquema de posicionamiento, [1](#)
- 'estáticos' grupo de medios, [1](#)
- estructura del documento, [1](#)
- estructura del formato, [1](#)
- ex (unidad), [1](#)

- first-child, [1](#)
- first-letter, [1](#)
- first-line, [1](#)
- 'float', [1](#)
- foco, [1](#)
- focus (pseudo-clase), [1](#)
- 'font', [1](#)
- 'font-family', [1](#)
- 'font-size', [1](#)
- 'font-style', [1](#)
- 'font-variant', [1](#)
- 'font-weight', [1](#)
- <forma>
 - definición de, [1](#)
- <frecuencia>, [1](#)
 - definición de, [1](#)
- fuentes del sistema, [1](#)

- georgian, [1](#)
- 'groove', [1](#), [2](#)
- grupo de medios, [1](#)

- 'height', [1](#)
- hermano, [1](#)
- herramientas de autor, [1](#)
- 'hidden', [1](#)
- 'hidden', [1](#)

- hijo, [1](#)
- hoja, [1](#)
- hoja de estilo, [1](#)
- hoja de estilo predeterminada, [1](#)
- hoja de estilo válida, [1](#)
- hover (pseudo-clase), [1](#)
- ícono sonoro, [1](#)
- identificador, [1](#)
- identificador, definición de, [1](#)
- ignorar, [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#)
- ilícito, [1](#)
- importación condicional, [1](#)
- importación dependiente de los medios, [1](#)
- inherit, definición de, [1](#)
- 'inline', definición de, [1](#)
- 'inline-block', definición de, [1](#)
- 'inset', [1](#), [2](#)
- 'interactivos' grupo de medios, [1](#)
- interlineado, [1](#)
- invert, [1](#)
- iso-10646, [1](#)

- LALR(1), [1](#)
- lang (pseudo-clase), [1](#)
- lector de pantalla, [1](#)
- 'left', [1](#)
- <left>
 - definición de, [1](#)
- lenguaje (humano), [1](#)
- lenguaje del documento, [1](#)
- 'letter-spacing', [1](#)
- lienzo, [1](#), [2](#)
- ligaduras, [1](#)
- límite del borde, [1](#)
- límite del contenido, [1](#)
- límite del margen, [1](#)
- límite del relleno, [1](#)
- límite externo, [1](#)
- límite interno, [1](#)
- 'line-height', [1](#)
- link (pseudo-clase), [1](#)
- 'list-item', definición de, [1](#)
- 'list-style', [1](#)
- 'list-style-image', [1](#)
- 'list-style-position', [1](#)
- 'list-style-type', [1](#)
- lower-greek, [1](#)

- lower-latin, [1](#)
- lower-roman, [1](#)

- margen
 - horizontal, [1](#)
 - de una caja, [1](#)
 - vertical, [1](#)
- margen vertical, [1](#)
- márgenes cerrados, [1](#)
- márgenes horizontales, [1](#)
- 'margin', [1](#)
- 'margin-bottom', [1](#)
- 'margin-left', [1](#)
- 'margin-right', [1](#)
- 'margin-top', [1](#)
- <margin-width>
 - definición de, [1](#)
- 'max-height', [1](#)
- 'max-width', [1](#)
- <medida>, [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#)
 - definición de, [1](#)
- medida absoluta, [1](#)
- medio interlineado, [1](#)
- medios, [1](#)
- 'min-height', [1](#)
- 'min-width', [1](#)
- modelo de formato visual, [1](#)

- nivel de pila, [1](#)
- NO DEBE, [1](#)
- NO DEBERÁ, [1](#)
- NO DEBERÍA, [1](#)
- no-close-quote, [1](#), [2](#)
- no-open-quote, [1](#), [2](#)
- 'none'
 - como estilo de borde, [1](#), [2](#)
 - como valor de display, [1](#)
- normal, [1](#)
- nueva línea, [1](#)
- numeración automática, [1](#)
- <número>, [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#)
 - definición de, [1](#)

- OPCIONAL, [1](#)
 - open-quote, [1](#), [2](#)
 - 'orphans', [1](#)
 - 'outline', [1](#)
 - 'outline-color', [1](#)
 - 'outline-style', [1](#)
 - 'outline-width', [1](#)
 - 'outset', [1](#), [2](#)
 - 'overflow', [1](#)
-
- 'padding', [1](#)
 - 'padding-bottom', [1](#)
 - 'padding-left', [1](#)
 - 'padding-right', [1](#)
 - 'padding-top', [1](#)
 - <padding-width>
 - definición de, [1](#)
 - padre, [1](#)
 - 'page-break-after', [1](#)
 - 'page-break-before', [1](#)
 - 'page-break-inside', [1](#)
 - 'paginados' grupo de medios, [1](#)
 - 'pause', [1](#)
 - 'pause-after', [1](#)
 - 'pause-before', [1](#)
 - 'pitch', [1](#)
 - 'pitch-range', [1](#)
 - pixel, [1](#)
 - pixel de referencia, [1](#)
 - 'play-during', [1](#)
 - <porcentaje>, [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#)
 - definición de, [1](#)
 - posicionamiento relativo, [1](#)
 - 'position', [1](#)
 - predeterminada
 - codificación de caracteres, [1](#)
 - 'property-name', [1](#)
 - propiedad, [1](#)
 - propiedad resumida, [1](#), [2](#), [3](#)
 - propiedades de lista, [1](#)
 - pseudo-clase
 - :first, [1](#)
 - :left, [1](#)
 - :right, [1](#)
 - pseudo-clases, [1](#)
 - :active, [1](#)
 - :focus, [1](#)
 - :hover, [1](#)

- :lang, [1](#)
 - :link, [1](#)
 - :visited, [1](#)
- pseudo-elementos, [1](#)
 - :after, [1](#), [2](#)
 - :before, [1](#), [2](#)
 - :first-letter, [1](#)
 - :first-line, [1](#), [2](#)
- PUEDE, [1](#)

- 'quotes', [1](#)

- raíz, [1](#)
- RECOMENDADO, [1](#)
- 'rejilla' grupo de medios, [1](#)
- regla-arroba, [1](#)
- reglas de los flotantes, [1](#)
- reglas-arroba, [1](#)
- Relleno
 - de una caja, [1](#)
- REQUERIDO, [1](#)
- Resource Identifier (URI), [1](#)
- 'richness', [1](#)
- 'ridge', [1](#), [2](#)
- 'right', [1](#)
- <right>
 - definición de, [1](#)
- run-in box, [1](#)
- 'run-in', definición de, [1](#)

- secuencia ficticia de marcas, [1](#), [2](#), [3](#)
- selector, [1](#), [2](#), [3](#), [4](#)
 - equivalente, [1](#)
 - sujeto del, [1](#)
- selector de hijo, [1](#)
- selector de página, [1](#)
- selector de tipo, [1](#)
- selector simple, [1](#)
- selector universal, [1](#)
- selectores de descendientes, [1](#)
- sensibilidad a mayúsculas/minúsculas, [1](#)
- sistema de reglas, [1](#)
- 'solid', [1](#), [2](#)
- 'speak', [1](#)

- 'speak-header', [1](#)
- 'speak-numeral', [1](#)
- 'speak-punctuation', [1](#)
- 'speech' grupo de medios, [1](#)
- 'speech-rate', [1](#)
- square, [1](#)
- 'stress', [1](#)
- sujeto (del selector), [1](#)

- tablas, [1](#)
- tabla en línea, [1](#)
- table, [1](#)
- table-caption, [1](#), [2](#), [3](#), [4](#), [5](#)
- table-cell, [1](#), [2](#), [3](#), [4](#), [5](#)
- table-column, [1](#)
- table-column-group, [1](#)
- table-footer-group, [1](#)
- table-header-group, [1](#)
- 'table-layout', [1](#)
- table-row, [1](#)
- table-row-group, [1](#)
- 'táctiles' grupo de medios, [1](#)
- <tamaño-absoluto>
 - definición de, [1](#)
- <tamaño-relativo>
 - definición de, [1](#)
- 'text-align', [1](#)
- 'text-decoration', [1](#)
- 'text-indent', [1](#)
- 'text-transform', [1](#)
- text/css, [1](#)
- <tiempo>, [1](#)
 - definición de, [1](#)
- tipo de medio, [1](#)
- 'todos' grupo de medios, [1](#)
- tokenizer, [1](#)
- 'top', [1](#)
- <top>
 - definición de, [1](#)

- unicode, [1](#)
- 'unicode-bidi', [1](#)
- unidad em (cuadratín), [1](#)
- unidades relativas, [1](#)
- Uniform Resource Locator (URL), [1](#)
- Uniform Resource Name (URN), [1](#)

- upper-latin, [1](#)
- upper-roman, [1](#)
- URI (Uniform Resource Identifier), [1](#)
- <uri>, [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
 - definición de, [1](#)
- URL (Uniform Resource Locator), [1](#)
- URN (Uniform Resource Name), [1](#)

- vacío, [1](#)
- valor, [1](#)
- valor computado, [1](#)
- valor especificado, [1](#)
- valor inicial, [1](#)
- valor real, [1](#)
- valor usado, [1](#)
- 'vertical-align', [1](#)
- 'visibility', [1](#)
- visited (pseudo-clase), [1](#)
- 'visuales' grupo de medios, [1](#)
- 'voice-family', [1](#)
- volume, [1](#)
- 'volume', [1](#)
- <voz-específica>
 - definición de, [1](#)
- <voz-genérica>, definición de, [1](#)

- 'white-space', [1](#)
- 'widows', [1](#)
- 'width', [1](#)
- 'word-spacing', [1](#)

- x-height, [1](#)

- 'z-index', [1](#)
- zona de recorte, [1](#)